dBase III PLUS

Novell NetWare kapcsolat

FoxBase+

CLIPPER

2

dBase III PLUS

Novell NetWare kapcsolat

FoxBase+

CLIPPER

2

ALKALMAZÁSTECHNIKAI TANÁCSADÓ SZOLGÁLAT

BUDAPEST, 1989

Szerkesztette: Szenes Katalin

Lektorálta: Szenes Katalin
Úry László

Készítette: Mondovics János *(1., 2., 5., 6., 8., 9., 10., 12. fejezet)*
Soltész Erzsébet *(3., 4., 7. fejezet)*
Cseh Zsolt *(11., 13. fejezet)*
Csaplár Ágnes *(14. fejezet)*
Úry László *(15. fejezet)*
Szenes Katalin *(1., 12. fejezet)*

Technikai szerkesztő: Nagy Olivér

Kiadó: LSI Alkalmazástechnikai Tanácsadó Szolgálat
Felelős kiadó: dr. Kovács Magda
Szedés: LSI ATSz Info osztály
Engedélyszám: 17078
ISBN I. kötet: 963 592 930 7
ISBN összkötet: 963 592 932 3
Készült: Bástya–Textprint
Felelős vezető: Balanyi István

Tartalomjegyzék

I. kötet

Előszó 3

1. Relációs adatbázisok alapfogalmai 6
   1.1. Az adatbázisok rövid fogalmi áttekintése 6
   1.2. Táblázatos ábrázolás 7
   1.3. Relációs adatbázisok 8
   1.4. A relációs adatbázisok előnyei 9

2. A dBASE III PLUS technikai jellemzői, alkalmazási korlátok és lehetőségek 10

3. File-ok, file-típusok, adattípusok 17
   3.1. File-típusok 17
      3.1.1. Adatfile (database)(.dbf) 17
      3.1.2. Feljegyzés file (database memo)(.dbt) 17
      3.1.3. Indexfile (.ndx) 17
      3.1.4. Szűrő, kereső file (query)(.qry) 18
      3.1.5. Jelentés szerkezeti file (report form)(.frm) 18
      3.1.6. Címke file (label)(.lbl) 18
      3.1.7. Szerkezeti file (report)(.fmt) 18
      3.1.8. Képernyő file (screen)(.scr) 19
      3.1.9. Katalógus file (catalog)(.cat) 19
      3.1.10. Memória file (memory)(.mem) 19
      3.1.11. Szöveg file (text)(.txt) 19
      3.1.12. Környezetet kijelölő file (view)(.vue) 19
      3.1.13. Programfile (.prg) 20
   3.2. Adatbázis struktúra 20
      3.2.1. A stuktúra elemei 20
      3.2.2. Az adatbázis struktúra megadásának menete 21
      3.2.3. Az adatbázis kezelését könnyítő utasítások és függvények 22
   3.3. Az adatbázis mezők típusai 23
   3.4. Memóriaváltozók 24
      3.4.1. A memóriaváltozók alkalmazásának feltételei 24
      3.4.2. A memóriaváltozókban tárolható adattípusok 25
      3.4.3. A memóriaváltozók használata 26

4. Programozás dBase III Plus-ban Assist-tel 27
   4.1. A Képernyő felépítése 28
   4.2. A menük és használatuk 31
      4.2.1. A menüválasztás menete (horizontális) 31
      4.2.2. Menüopció választás (vertikális) 31

4.3. A menüopciók funkciója, használata 33
4.3.1. Adatfile létrehozása 33
4.3.2. Rekord hozzáfűzése az adatfile-hoz 36
4.3.3. Az adatmegjelenítés formátumának meghatározása 39
4.3.3.1. A menük funkciói vázlatosan 39
4.3.3.2. Mezők kijelölése 40
4.3.3.3. Üres sorok és karakterek beszúrása, törlése 42
4.3.3.4. Szöveg elhelyezése a képernyőn 43
4.3.3.5. Mezők mozgatása 44
4.3.3.6. Mezőnevek megváltoztatása 45
4.3.3.7. A mező hosszának megváltoztatása 46
4.3.3.8. Az adatbeviteli formátum kiegészítése új mezővel 47
4.3.3.9. Mező törlése a formátumból 48
4.3.3.10. A mezők bemeneti adataira vonatkozó előírások megváltoztatása 48
4.3.3.11. Adatbevitel maszk (előírás a mező tartalmára) 49
4.3.3.12. Keretezés a képernyőn 50
4.3.3.13. A képernyőformátum kinyomtatása 51
4.3.3.14. A formátum mentése 51
4.3.4. Adatmegjelenítési lehetőségek 51
4.3.5. Rekord módosítása 54
4.3.6. Rekord törlése 55
4.3.7. Pozícionálás az adatfile-ban 56
4.3.7.1. Pozícionálás rekordsorszám szerint 57
4.3.7.2. Rekord megkeresése a mező tartalma alapján 58
4.3.7.3. Rekordcsoportok kiíratása 59
4.3.8. Adatrendezés 61
4.3.8.1. Rekordok fizikai rendezése 61
4.3.8.2. Indexfile létrehozása 61
4.3.8.3. Indexfile aktivizálása, karbantartása 62
4.3.8.4. Keresés az indexfile segítségével 63
4.3.9. Új adatfile létrehozása egy meglévőből 64
4.3.10. Szűrőfeltételek felállítása (a"query" file használata) 64
4.3.10.1. Szűrőfeltételek megadása ("query" file létrehozása) 65
4.3.10.2. A feltételek egymásba ágyazása 68
4.3.10.3. A szűrőfeltétel ellenőrzése, korrigálása 69
4.3.10.4. A szűrőfile használata 70
4.3.11. Címke és nyomtatási formátum file-ok használata 70
4.3.12. File-ok és kapcsolatuk nyilvántartása "Assist"-tal 72
4.3.12.1. A környezeti (view) file 72
4.3.12.2. A katalógusfile használata 76
4.3.13. Kapcsolat létesítése más adatkezelő programokkal 76

5. Editor, vezérlőbillentyűk 79

6. dBASE III PLUS módszertan 81
6.1. A HISTORY alkalmazása 81
6.2. File-műveletek 82
6.2.1. Megnyitás, lezárás 82
6.2.2. Tartalomtervezés 85

6.2.3. Tartalom megváltoztatása 86
6.2.4. Adatkeresés 89
6.2.5. Több file egyidejű használata 89
6.2.6. Képernyő formátum file-ok 91
6.2.7. Adatkonverzió 93
6.3. A nyomtató vezérlése 94
6.3.1. Vezérlés dBASE III PLUS parancsokkal 94
6.3.2. Vezérlő kódok 95
6.3.3. Különféle nyomtatásvezérlési módok 96
6.4. Adatkatalógus 97
6.4.1. File-ok szervezése 97
6.4.2. Adatbázis előállítása 98
6.4.3. Katalógusok előállítása, aktivizálása 98
6.4.4. Műveletek a katalóguson 99

7. A dBASE III PLUS és a dBASE III összevetése 101
7.1. Újdonságok a formátumtervezésben és az adatmegjelenítésben 101
7.1.1. A képernyő formátum tervezése, a képernyő-file 102
7.1.2. Az adatmegjelenítési lehetőségek bővülése 102
7.2. Feltételszűkítés, szűrőfeltételek 104
7.3. Adatfile-ok rendezettségével kapcsolatos kiegészítések 105
7.4. Új lehetőség adatfile létrehozására 106
7.5. File-ok és kapcsolatuk nyilvántartása 106
7.6. A dBASE III PLUS együttműködése más programokkal 107
7.6.1. A dBASE III PLUS kapcsolata más adatkezelőkkel 107
7.6.2. Más programok hívása dBASE III PLUS-ból 108
7.7. Új utasítások a párbeszédes kapcsolatban 109
7.8. Új lehetőségek a program és környezete kapcsolatában 109
7.9. Új lehetőségek a tesztelésben és a hibajavításban 110
7.9.1. A parancsarchiválás 110
7.9.2. Programozott eseményfigyelés 111
7.9.3. Hibajavítást segítő függvények 111
7.10. Szintaktikai kiegészítések a dBASE III-hoz képest 112

8. A dBASE III PLUS belső struktúrája 113
8.1. Adatbázis- (.dbf) file header struktúrája 113
8.2. Az adatrekordok szerkezete 113
8.3. Memóriafile-ok struktúrája 114
8.4. File eleje/vége feltételek 115
8.4.1. Indexeletlen file-ok 115
8.4.2. Indexelt file-ok 116
8.4.3. File-t (szűrő) feltételek 117

9. Függvények 118
9.1. Input és output adattípusok 118
9.2. A függvények osztályozása 120
9.3. A függvények részletes leírása 122

9.3.1. &-makro funkció 122
9.3.2. ABS 122
9.3.3. ASC 122
9.3.4. AT 122
9.3.5. BOF 122
9.3.6. CDOW 123
9.3.7. CHR 123
9.3.8. CMONTH 123
9.3.9. COL 123
9.3.10. CTOD 124
9.3.11. DATE 124
9.3.12. DAY 124
9.3.13. DBF 124
9.3.14. DELETED 125
9.3.15. DISKSPACE 125
9.3.16. DOW 125
9.3.17. DTOC 125
9.3.18. EOF 125
9.3.19. ERROR 126
9.3.29. EXP 126
9.3.21. FIELD 126
9.3.22. FILE 127
9.3.23. FKLABEL 127
9.3.24. FKMAX 127
9.3.25. FOUND 127
9.3.26. GETENV 128
9.3.27. IIF 128
9.3.28. INKEY 128
9.3.29. INT 129
9.3.30. ISALPHA 129
9.3.31. ISCOLOR 130
9.3.32. ISOLWER 130
9.3.33. ISUPPER 130
9.3.34. LEFT 130
9.3.35. LEN 131
9.3.36. LOG 131
9.3.37. LOWER 131
9.3.38. LTRIM 131
9.3.39. LUPDATE 131
9.3.40. MAX 132
9.3.41. MESSAGE 132
9.3.42. MIN 132
9.3.43. MOD 132
9.3.44. MONTH 133
9.3.45. NDX 133
9.3.46. OS 133
9.3.47. PCOL 134
9.3.48. PROW 134
9.3.49. READKEY 134
9.3.50. RECCOUNT 136

9.3.51. RECNO 136
9.3.52. RECSIZE 137
9.3.53. REPLICATE 137
9.3.54. RIGHT 138
9.3.55. ROUND 138
9.3.56. ROW 138
9.3.57. RTRIM 138
9.3.58. SPACE 139
9.3.59. SQRT 139
9.3.60. STR 139
9.3.61. STUFF 139
9.3.62. SUBSTR 140
9.3.63. TIME 140
9.3.64. TRANSFORM 140
9.3.65. TRIM 141
9.3.66. TYPE 141
9.3.67. UPPER 142
9.3.68. VAL 142
9.3.69. VERSION 142
9.3.70. YEAR 142

10. A dBASE III PLUS parancsok leírása 143

10.1. A parancsok funkcionális csoportosítása 143
10.1.1. File-létrehozó parancsok 143
10.1.2. Adatbeviteli parancsok 143
10.1.3. Adatszerkesztő parancsok 143
10.1.4. Felhasználói támogatás 144
10.1.5. Az adatok megjelenítése 144
10.1.6. Rekordpointer-pozícionálás 145
10.1.7. Adatbázisok kezelése 145
10.1.8. Más típusú file-ok kezelése 145
10.1.9. A memóriaváltozók használata 146
10.1.10. Programozás 146
10.1.11. Környezeti parancsok 147
10.1.12. Paramétervezérlések 147
10.1.13. Külső programkapcsolódások 149
10.1.14. Nyomkövető parancsok 149
10.1.15. Eseményfeldolgozás 150
10.2. A parancsok szintaktikus leírása 151
10.2.1. ?/?? 151
10.2.2. @ 151
10.2.3. @...TO 154
10.2.4. ACCEPT 154
10.2.5. APPEND 155
10.2.6. APPEND FROM 156
10.2.7. ASSIST 157
10.2.8. AVERAGE 157
10.2.9. BROWSE 158
10.2.10. CALL 159
10.2.11. CANCEL 160

10.2.12. CHANGE 160
10.2.13. CLEAR 161
10.2.14. CLEAR ALL 161
10.2.15. CLEAR FIELDS 161
10.2.16. CLEAR GETS 161
10.2.17. CLEAR MEMORY 162
10.2.18. CLEAR TYPEAHEAD 162
10.2.19. CLOSE 162
10.2.20. CONTINUE 162
10.2.21. COPY 163
10.2.22. COPY FILE 164
10.2.23. COPY STRUCTURE 165
10.2.24. COPY STRUCTURE EXTENDED 165
10.2.25. COUNT 165
10.2.26. CREATE 165
10.2.27. CREATE FROM 166
10.2.28. CREATE/MODIFY LABEL 166
10.2.29. CREATE/MODIFY QUERY 168
10.2.30. CREATE/MODIFY REPORT 170
10.2.31. CREATE/MODIFY SCREEN 173
10.2.32. CREATE/MODIFY VIEW 175
10.2.33. CREATE VIEW FROM ENVIRONMENT 177
10.2.34. DELETE 177
10.2.35. DIR 178
10.2.36. DISPLAY 178
10.2.37. DISPLAY HISTORY 179
10.2.38. DISPLAY MEMORY 179
10.2.39. DISPLAY STATUS 180
10.2.40. DISPLAY STRUCTURE 180
10.2.41. DO 181
10.2.42. DO CASE 181
10.2.43. DO WHILE 182
10.2.44. EDIT 183
10.2.45. EJECT 183
10.2.46. ERASE 183
10.2.47. EXIT 184
10.2.48. EXPORT 184
10.2.49. FIND 185
10.2.50. GO/GOTO 185
10.2.51. HELP 186
10.2.52. IF 186
10.2.53. IMPORT 186
10.2.54. INDEX 187
10.2.55. INPUT 188
10.2.56. INSERT 188
10.2.57. JOIN 189
10.2.58. LABEL 190
10.2.59. LIST 190
10.2.60. LIST HISTORY 190
10.2.61. LIST MEMORY 191

10.2.62. LIST STATUS 191
10.2.63. LIST STRUCTURE 191
10.2.64. LOAD 191
10.2.65. LOCATE 192
10.2.66. LOOP 193
10.2.67. MODIFY COMMAND 193
10.2.68. MODIFY LABEL 194
10.2.69. MODIFY QUERY 194
10.2.70. MODIFY REPORT 194
10.2.71. MODIFY SCREEN 194
10.2.72. MODIFY STRUCTURE 195
10.2.73. MODIFY VIEW 196
10.2.74. NOTE/* 196
10.2.75. ON 196
10.2.76. PACK 197
10.2.77. PARAMETERS 197
10.2.78. PRIVATE 198
10.2.79. PROCEDURE 198
10.2.80. PUBLIC 198
10.2.81. QUIT 199
10.2.82. READ 199
10.2.83. RECALL 200
10.2.84. REINDEX 200
10.2.85. RELEASE 200
10.2.86. RENAME 200
10.2.87. REPLACE 201
10.2.88. REPORT 201
10.2.89. RESTORE 202
10.2.90. RESUME 203
10.2.91. RETRY 203
10.2.92. RETURN 203
10.2.93. RUN/! 204
10.2.94. SAVE 204
10.2.95. SEEK 205
10.2.96. SELECT 205
10.2.97. SET 206
10.2.98. SET ALTERNATE 207
10.2.99. SET BELL 208
10.2.100. SET CARRY 208
10.2.101. SET CATALOG 208
10.2.102. SET CATALOG TO 209
10.2.103. SET CENTURY 209
10.2.104. SET COLOR 210
10.2.105. SET CONFIRM 211
10.2.106. SET CONSOLE 211
10.2.107. SET DATE 211
10.2.108. SET DEBUG 212
10.2.109. SET DECIMAL 212
10.2.110. SET DEFAULT 212
10.2.111. SET DELETED 213

10.2.112. SET DELIMITERS 213
10.2.113. SET DEVICE 213
10.2.114. SET DOHISTORY 214
10.2.115. SET ECHO 214
10.2.116. SET ESCAPE 215
10.2.117. SET EXACT 215
10.2.118. SET FIELDS 215
10.2.119. SET FIELDS TO 216
10.2.120. SET FILTER 219
10.2.121. SET FIXED 220
10.2.122. SET FORMAT 220
10.2.123. SET FUNCTION 221
10.2.124. SET HEADING 221
10.2.125. SET HELP 221
10.2.126. SET HISTORY 222
10.2.127. SET HISTORY TO 222
10.2.128. SET INDEX 222
10.2.129. SET INTENSITY 223
10.2.130. SET MARGIN 223
10.2.131. SET MEMOWIDTH TO 223
10.2.132. SET MENU 224
10.2.133. SET MESSAGE TO 224
10.2.134. SET ORDER 224
10.2.135. SET PATH 225
10.2.136. SET PRINT 225
10.2.137. SET PRINTER 226
10.2.138. SET PROCEDURE 226
10.2.139. SET RELATION 226
10.2.140. SET SAFETY 227
10.2.141. SET SCOREBOARD 228
10.2.142. SET STATUS 228
10.2.143. SET STEP 229
10.2.144. SET TALK 229
10.2.145. SET TITLE 229
10.2.146. SET TYPEAHEAD TO 230
10.2.147. SET UNIQUE 230
10.2.148. SET VIEW TO 231
10.2.149. SKIP 231
10.2.150. SORT 232
10.2.151. STORE 232
10.2.152. SUM 233
10.2.153. SUSPEND 233
10.2.154. TEXT 234
10.2.155. TOTAL 234
10.2.156. TYPE 235
10.2.157. UPDATE 235
10.2.158. USE 235
10.2.159. WAIT 236
10.2.160. ZAP 236

II. kötet

11. dBASE példaprogram 238

12. A hibaüzenetek, rövid magyarázattal 250

13. A dBASE III PLUS és a lokális hálózatok 284

13.1. Lokális hálózatok 284
13.1.1. A lokális hálózatok és a dBASE III PLUS 285
13.2. A dBASE III PLUS hálózati programcsomag 286
13.2.1. A dBASE III PLUS programcsomag elemei 286
13.2.2. Telepítési követelmények 286
13.2.3. A dBASE ADMINISTRATOR telepítése 287
13.2.4. A dBASE ACCESS telepítése 288
13.3. Adatvédelem, adatvédelmi szintek, dBASE adatvédelem 289
13.3.1. Novell/Netware LAN hálózati adatvédelem 289
13.3.1.1. A hálózatba kapcsolódás 289
13.3.1.2. Könyvtárhozzáférés 289
13.3.2. dBASE III PLUS adatvédelem – alapfogalmak 291
13.3.2.1. A bejelentkezési védelem és a DBSYSTEM.DB file 291
13.3.2.1.1. A bejelentkezés védelem 291
13.3.2.1.2. A DBSYSTEM.DB file 292
13.3.2.2. Adatkódolás (titkosítás) és dekódolás 292
13.3.2.2.1. A kódolás (titkosítás) kulcsa és a csoport (group) fogalma 292
13.3.2.2.2. A SET ENCRYPTION utasítás 293
13.3.2.3. File és mezőhozzáférés 294
13.3.2.3.1. File előjogok 294
13.3.2.3.2. Mező előjogok 294
13.3.2.3.3. A titkosítás és az előjogok 296
13.3.3. A védelmi rendszer létrehozása, a PROTECT használata 296
13.3.3.1. PROTECT használati alapismeretek 296
13.3.3.1.1. A PROTECT indítása, adminisztrátor jelszó meghatározása 296
13.3.3.1.2. A PROTECT képernyőszerkezete 297
13.3.3.2. A PROTECT használata 299
13.3.3.2.1. Profil létrehozása 299
13.3.3.2.2. File és mező hozzáférési előjogszintek meghatározása 301
13.3.3.2.3. Profilok és előjogok elmentése, kilépés 306
13.4. Programozás hálózati környezetben 309
13.4.1. Az adatsérthetetlenség védelme 309
13.4.1.1. Zárolás a dBASE III PLUS-ban 310
13.4.1.2. A dBASE III PLUS zárolási eszköztára 311
13.4.1.2.1. Teljes képernyős rekord megjelenítési, módosítási parancsok 311
13.4.1.2.2. File/rekord zárolási parancsok 312
13.4.1.2.3. File-zárolás 312
13.4.1.2.4. Rekord-zárolás 312
13.4.1.2.5. File/rekord zárolás időtartama 313
13.4.1.2.6. Több file zárolása 313
13.4.1.2.7. Több rekord zárolása 313

13.4.1.3. A file megnyitás és a hozzáférés 314
13.4.2. A holtpont és elkerülése 317
13.4.3. Hibák észlelése és az "újraélesztés" 318
13.4.4. Adatvédelem a programozás szintjén 319
13.4.5. Példaprogram 319
13.5. Hálózati parancs és függvénygyűjtemény 329
13.5.1. Felhasználót segítő információkat szolgáltató parancsok 329
13.5.1.1. DISPLAY STATUS (TO PRINT) 329
13.5.1.2. LIST STATUS (TO PRINT) 330
13.5.1.3. DISPLAY USERS 330
13.5.2. Adatbázis védelem 331
13.5.2.1. SET ENCRYPTION ON/OFF 331
13.5.2.2. LOGOUT 332
13.5.3. Paraméter beállítás 332
13.5.3.1. SET EXCLUSIVE ON/OFF 332
13.5.3.2. SET PRINTER TO 333
13.5.4. Adatbázis kezelés 333
13.5.4.1. USE 333
13.5.5. Zárolással kapcsolatos parancsok és függvények 334
13.5.5.1. FLOCK() 334
13.5.5.2. RLOCK()/LOCK() 334
13.5.5.3. UNLOCK (ALL) 334
13.5.6. Más, programozást segítő parancsok és függvények 335
13.5.6.1. RETRY 335
13.5.6.2. ACCESS() 335
13.5.6.3. ERROR() 335
13.5.6.4. MESSAGE() 335

14. CLIPPER 336

14.1. A dBASE III PLUS és a Clipper 336
14.1.1. A Clipper, mint adatbáziskezelő programcsomag 336
14.1.2. A Clipper, mint dBASE III PLUS fordító 336
14.1.3. Az interpreter és a fordító közötti különbségek 336
14.1.4. A Clipper előnyei 337
14.1.5. A gépi környezettel szemben támasztott követelmények 338
14.1.5.1. Hardware feltételek 338
14.1.5.2. A DOS-környezet módosításai 338
14.1.5.2.1. A DOS parancs-processzor 338
14.1.5.2.2. A DOS FILES és BUFFERS paraméterei 339
14.1.5.2.3. Az ANSI terminál támogatás 339
14.2. Fordítás és szerkesztés 340
14.2.1. Fordítás 340
14.2.1.1. A Clipper fordító speciális futási paraméterei 341
14.2.1.1.1. Bővített eredeti kód (-n) 341
14.2.1.1.2. Sorszámok nélkül (-l) 341
14.2.1.1.3. A program szintaktikus ellenőrzése (-s) 341
14.2.1.2. A név.CLP file-ok használata 341
14.2.2. Szerkesztés 342
14.2.2.1. A PLINK86 parancsai 343
14.2.2.1.1. # 343

14.2.2.1.2. BEGINAREA 343
14.2.2.1.3. DEBUG 343
14.2.2.1.4. ENDAREA 343
14.2.2.1.5. FILE 343
14.2.2.1.6. LIBRARY 343
14.2.2.1.7. MAP 344
14.2.2.1.8. OVERLAY 344
14.2.2.1.9. OUTPUT 344
14.2.2.1.10. SECTION 344
14.2.2.1.11. SECTION INTO 344
14.2.2.1.12. VERBOSE 344
14.2.2.2. A PLINK86 hívása egyetlen parancssorban 345
14.2.2.3. Interaktív üzemmód 345
14.2.2.4. A név.LNK file használata 345
14.2.2.5. Overlay-szervezés 346
14.3. A Clipper-t alkotó file-ok és használatuk 348
14.3.1. Végrehajtható típusú (.EXE) file-ok 348
14.3.2. Könyvtárak (.LIB) 348
14.3.3. Egyéb kiegészítő file-ok 348
14.4. Adatbáziskezelés, kompatibilitási problémák a dBASE III PLUS és a Clipper között 350
14.4.1. A Clipper által nem támogatott dBASE III PLUS parancsok 350
14.4.2. A Clipper újdonságai a dBASE III PLUS-hoz képest 351
14.4.2.1. @...BOX 352
14.4.2.2. @...SAY...GET 352
14.4.2.3. APPEND FROM 352
14.4.2.4. AVERAGE 353
14.4.2.5. CALL 353
14.4.2.6. COUNT 353
14.4.2.7. CREATE 353
14.4.2.8. DECLARE 354
14.4.2.9. DISPLAY 355
14.4.2.10. EXTERNAL 355
14.4.2.11. FOR...NEXT 356
14.4.2.12. FUNCTION...RETURN 356
14.4.2.13. KEYBOARD 356
14.4.2.14. LIST 357
14.4.2.15. PRIVATE 357
14.4.2.16. PROCEDURE 357
14.4.2.17. RESTORE SCREEN 358
14.4.2.18. SAVE SCREEN 358
14.4.2.19. SET COLOR TO 358
14.4.2.20. SET ESCAPE 359
14.4.2.21. SET FUNCTION 359
14.4.2.22. SET KEY 359
14.4.2.23. SET MESSAGE TO...@...PROMPT...MENU TO 360
14.4.2.24. SET RELATION TO 361
14.4.2.25. SUM 361
14.4.2.26. && 361
14.4.2.27. DTOS() 362

14.4.2.28. EMPTY() 362
14.4.2.29. FIELDNAME() 362
14.4.2.30. IF() 363
14.4.2.31. INKEY() 363
14.4.2.32. LASTKEY() 363
14.4.2.33. LASTREC() 364
14.4.2.34. MEMOEDIT() 364
14.4.2.35. PROCLINE() 364
14.4.2.36. PROCNAME() 365
14.4.2.37. REPLICATE() 365
14.4.2.38. SECONDS() 365
14.4.2.39. SELECT() 365
14.4.2.40. UPDATED() 366
14.4.3. Működési különbségek 366
14.4.3.1. Memóriamezők fejlettebb kezelése 366
14.4.3.2. A makro-szerkezetek új lehetőségei 366
14.4.3.3. Új operátorok 366
14.4.3.4. Index file-ok 367
14.4.3.5. Kurzorvezérlő billentyűk 367
14.4.4. Nyomkövetés, tesztelés (a Clipper Debugger) 367
14.4.5. Az inkompatibilitások lehetséges feloldásai 369
14.4.5.1. A Clipper PUBLIC változó használata 369
14.4.5.2. A Clipper Utility programjainak használata 370
14.4.5.3. A Clipper bővített könyvtárának használata 371
14.5. Parancsok és funkciók összefoglalása 372
14.6. Hibaüzenetek 384
14.6.1. Fordítás 384
14.6.2. Szerkesztés 387
14.6.3. Futás közben kiadott hibaüzenetek 391
14.7. A '86. őszi változat (Többfelhasználós környezet) 392
14.8. A '87. nyári változat 395
14.9. Példa 400

15. A FoxBase+ fejlesztő környezet 414

15.1. A FoxBase+ rendszer sajátosságai 414
15.1.1. Bevezetés 414
15.1.2. FoxBase+ bővítések 414
15.1.2.1. Tömbváltozók használata 414
15.1.2.2. Többszörös RELATION használata 415
15.1.2.3. Rendszerfüggvények 415
15.1.2.4. Eljárások száma az eljárásfile-ban 415
15.1.2.5. Eljárásfile-ok létrehozása 415
15.1.2.6. A dBASE II file-kompatibilitás 415
15.1.2.7. MIN/MAX függvények a dátumokon 415
15.1.2.8. A DTOC függvények módosítása 415
15.1.2.9. BROWSE utasítás módosítása 416
15.1.2.10. MODIFY COMMAND utasítás módosítása 416
15.1.2.11. Nyitott file-ok száma 416
15.1.2.12. Nyomtató időtúllépése 416

15.1.2.13. Felhasználói függvények 416
15.1.3. Nem támogatott elemek 417
15.1.4. A FoxBase + hardware igénye és technikai adatai 418
15.2. A FoxBase + programrendszer felépítése 418
15.2.1. A fordító 418
15.2.2. Eljárás-file generálása 419
15.2.3. A FoxBase + interpretere 419
15.2.4. A futtató rendszer 420
15.2.5. Képernyő kezelés 420
15.3. A FoxBase + rendszer konfigurálása 420
15.3.1. Bevezetés 420
15.3.2. Konfigurációs tételek 421
15.3.2.1. BUCKET 421
15.3.2.2. BUFFERS 421
15.3.2.3. FILES 421
15.3.2.4. HISTORY 421
15.3.2.5. HMEMORY 421
15.3.2.6. INDEX 422
15.3.2.7. MAXMEM 422
15.3.2.8. MVARSIZ 422
15.3.2.9. MVCOUNT 422
15.3.2.10. PCACHE 422
15.3.2.11. PROMPT 422
15.3.2.12. TEDIT 422
15.3.2.13. TIME 422
15.3.2.14. WP 423
15.3.3. A használható konfigurációs tételek összefoglalása 423
15.4. Többfelhasználós rendszer 424
15.4.1. Bevezetés 424
15.4.2. Környezeti feltételek 424
15.4.3. A többfelhasználós FoxBase + indítása 424
15.5. A SYS függvény 425

Index 427

Irodalomjegyzék 430

Hivatkozott márkanevek 431

11. dBASE példaprogram

```
*************************************************************
*    Program : Személyi adatok nyilvántartása               *
*                                                           *
*              - Felvétel                                   *
*              - Módositás                                  *
*              - Törlés                                     *
*              - Megjelenitès (képernyön és nyomtatón)      *
*                                                           *
*************************************************************
*                  Javasoljuk Önnek                         *
* Az ISKOLA.DBF és BEOSZTAS.DBF kódfile-ok karbantar-       *
* tásának programozását                                     *
************************************************************* *
* Kapcsoló beállitások

set safety off
set talk off
set status off
set bell off
set confirm on
* file ellenörzések --> létezik e mindegyik szükséges file ?"
clear
@ 8,8 to 12,50
@ 10,10 say "File ellenörzések ... "
@ 11,10 say "SZEMELY.DBF  ISKOLA.DBF  BEOSZTAS.DBF"
*
*Structure for database: C:szemely.dbf  -> adatbázis file struktura
*Number of data records:       1        -> rekordszám
*Date of last update    : 11/18/88      -> utolsó változtatás dátuma
*Field  Field Name  Type                 Dec
*    1  SZEZAM      Character     13          -> Személyi szám
*    2  NEV         Character     30          -> Név
*    3  CIM         Character     30          -> Lakcim
*    4  ISKOLA      Character      2          -> Iskolai végzettségkód
*    5  BEO         Character      2          -> Beosztáskód
*** Total **                      78
*
*Structure for database: C:iskola.dbf   -> adatbázis file struktura
*Number of data records:       3        -> rekordszám
*Date of last update    : 11/18/88      -> utolsó változtatás dátuma
*Field  Field Name  Type       Width     Dec
*    1  ISKOLA      Character      2           -> Iskolai végzettségkód
*    2  INEV        Character     20           -> Végzettség megnevezése
*** Total **                      23
*
*Structure for database: C:beosztas.dbf -> adatbázis file struktura
*Number of data records:       3        -> rekordszám
*Date of last update    : 11/18/88      -> utolsó változtatás dátuma
*Field  Field Name  Type       Width     Dec
*    1  BEO         Character      2           -> Beosztáskód
*    2  BNEV        Character     20           -> Beosztás megnevezése
*** Total **                      23
*
```

```
if .not.(file("SZEMELY.DBF").and.file("ISKOLA.DBF").and.file("BEOSZTAS.DBF"))
  clear
  @ 10,20 say " A program müködéséhez szükséges adatfile-ok :
  @ 12,20 say "   SZEMELY.DBF , ISKOLA.DBF, BEOSZTAS.DBF   "
  @  23,30 say "Üssön le egy billentyüt !!"
  set console off
  wait
  set console on
  cancel
endif

if .not.file("SZEMELY.NDX")
   use SZEMELY
   index on szezam to szemely
endif

if .not.file("ISKOLA.NDX")
   use ISKOLA
   index on iskola to ISKOLA
endif

if .not.file("BEOSZTAS")
   use BEOSZTAS
   index on beo to beosztas
endif

if .not.file("szemely.ndx")
   index on szezam to szemely
endif

do while .t.
   clear
   clear all

*  Keret rajzolás
   @  1,4 to 21,76
   @  22,0 to 24,79


   @  3,18 say "          Személyi adatok"
   @  4,18 say "----------------------------------"
   @  6,15 say "Választási lehetöségek :"
   @  8,22 say "1. Uj Személy felvétele"
   @ 10,22 say "2. Adott Személy adatainak módositása"
   @ 12,22 say "3. Adott Személy adatainak törlése"
   @ 14,22 say "4. Adott személy adatainak megjelenitése/listázása"
   @ 16,22 say "5. Kódfile-ok karbantartása"
   @ 18,22 say "0. KILÉPÉS"
   @ 20,15 say "Kérem válasszon :"
   VALASZ=0
   @ 20,43 get VALASZ picture "9" range 0,5
   read
   do case
     case VALASZ=0
*      Kilépés
       CLEAR ALL
       CLEAR
```

```
       @ 10,25 SAY "Program vége"
       CANCEL
     case  VALASZ=1

*  Személy adatainak felvétele :  SZEMELY.DBF adatfile
*  - személyi szám                SZEMELY.NDX indexfile személyi szám szerint
*  - név
*  - lakcim
*  - iskolai végzettség (kódolt) -> ISKOLA.DBF  ISKOLA.NDX végzettég kód szerint
*  - beosztás (kódolt)           -> BEOSZTAS.DBF BEOSZTAS.NDX beosztás kód
szerint

       do FELVESZ

     case VALASZ=2

*  Személy adatainak módositása :
*  személyi szám  bekérése
*  adatmódositás
*   - név
*   - lakcim
*   - iskolai végzettség (kódolt)
*   - beosztás (kódolt)

       do MODOSIT

    case VALASZ=3

*  Személy adatainak törlése
*  személyi szám  bekérése
*  rekordtörlés

       do TOROL

    case VALASZ=4

*  Személy adatainak megjelenitése
*  képernyön vagy nyomtatón

       do MEGJELEN


    case VALASZ=5

        clear

        @  10,20 say " Az ISKOLA.DBF és BEOSZTAS.DBF"
        @  12,20 say " file-ok karbantartása : felvétel, módositás, törlés"
        @  14,20 say " Önre vár !!"
        @  23,30 say "Üssön le egy billentyüt !!"
        set console off
        wait
        set console on
  endcase
enddo
```

FELVESZ.PRG

```
*  Személy adatainak felvétele :  SZEMELY.DBF adatfile
*  - személyi szám                SZEMELY.NDX indexfile személyi szám szerint
*  - név
*  - lakcim
*  - iskolai végzettség (kódolt) -> ISKOLA.DBF  ISKOLA.NDX végzettég kód szerint
*  - beosztás (kódolt)           -> BEOSZTAS.DBF BEOSZTAS.NDX beosztás kód
szerint



clear
@  1,4 to 21,76
@  22,0 to 24,79
@  3,18 say "       Személyi adatok felvétele"
@  4,18 say "-------------------------------------"

* adat és indexfile-ok megnyitása
select 1
use szemely index szemely alias SZEM
select 2
use iskola index iskola   alias ISK
select 3
use beosztas index beosztas  alias BEO


OK_1=.t.

do while OK_1
  v_szezam=space(13)
  v_nev=space(30)
  v_cim=space(30)
  v_iskola=space(2)
  v_beo=space(2)
  OK_2=.t.
  do while OK_2
    @ 8,10  say "Személyi szám           : " get v_szezam picture "9-999999-
9999"
    @ 10,10 say "Név                     : " get v_nev
    @ 12,10 say "Lakcim                  : " get v_cim
    @ 14,10 say "Iskolai végzettség (kód) : " get v_iskola picture "99"
    @ 16,10 say "Beosztás (kód)          : " get v_beo    picture "99"
    read
* utánanézünk hogy az adott személyi számmal
* rendelkezö személy létezik e már adatállományunkban
*
    sele SZEM
    seek v_szezam
    if eof()

* Iskolai végzettségkód alapján
* a megfelelö megnevezős kikeresése
* és kiirása a képernyöre
*
      sele ISK
```

```
      seek v_iskola
      if .not.eof()
        v_inev=inev
      else
        v_inev="Ismeretlen"
      endif

      @ 14,50 say space(20)
      @ 14,50 say v_inev
*
* Beosztáskód alapján
* a megfelelö megnevezŐs kikeresése
* és kiirása a képernyöre
*
      sele BEO
      seek v_beo
      if .not.eof()
        v_bnev=bnev
      else
        v_bnev="Ismeretlen"
      endif

      @ 16,50 say space(20)
      @ 16,50 say v_bnev
*
      @ 23,2 say space(77)
      @ 23,20 say "Tárolható ez az adat ? (i,n)"
      set console off
      wait to ignem
      set console on

      if upper(ignem)="I"

*  Uj rekord tárolása
*  a SZEMELY.DBF file-ban
*
        sele SZEM
        append blank
        replace szezam with v_szezam
        replace nev with v_nev
        replace cim with cim
        replace iskola with v_iskola
        replace beo with v_beo
        OK_2=.f.
*
      else
        @ 23,2 say space(77)
        @ 23,20 say "Kiván javitani ? (i,n)"
        set console off
        wait to ignem
        set console on
        @ 23,2 say space(77)
        OK_2=(upper(ignem)="I")
      endif
    else
      @ 23,2 say space(77)
```

```
      @ 23,10 say "Ez a személy már szerepel állományunkban ! Kiván javitani ?
(i,n)"
      set console off
      wait to ignem
      set console on
      @ 23,2 say space(77)
      OK_2=(upper(ignem)="I")
   endif
  enddo
  @ 23,2 say space(77)
  @ 23,20 say "Ujabb személy felvétele ? (i,n)"
  set console off
  wait to ignem
  set console on
  OK_1=(upper(ignem)="I")
  @ 23,2 say space(77)
enddo
close data
return
```

MEGJELEN.PRG

```
*  Személy adatainak törlése :
*  személyi szám bekérése
*  rekordtörlés

clear
@  1,4 to 21,76
@  22,0 to 24,79
@  3,18 say "     Személyi adatok megjelenitése"
@  4,18 say "-------------------------------------"

* adat és indexfile-ok megnyitása

select 1
use szemely index szemely alias SZEM
select 2
use iskola index iskola   alias ISK
select 3
use beosztas index beosztas  alias BEO
sele SZEM

* Az iskola és beosztásmegnevezés
* elökeresése a személykereséssel
* egyidöben --> SET REALTION

set relation to iskola into ISK
set relation to beo into BEO

OK_1=.t.
v_szezam=space(13)

do while OK_1
  @ 8,10  say "Személyi szám           : " get v_szezam picture "9-999999-9999"
```

```
  read
* utánanézünk hogy az adott személyi számmal
* rendelkezö személy létezik e már adatállományunkban
*
  sele SZEM
  seek v_szezam
  if .not.eof()

    v_nev=nev
    v_cim=cim
    v_iskola=iskola
    v_beo=beo

* a SET RELATION parancs révén
* az ISKOLA file és a BEOSZTAS file mutatója is a SZEMELY.DBF
* ISKOLA és BEO mezöinek megfelelö
* rekordra mutat
*
     v_inev=ISK -> inev
     v_bnev=BEO -> bnev
     valasz=1
     @ 4,55 to 8,75
     @ 5,57 say "1.Képernyö"
     @ 6,57 say "2.Nyomtato"
     @ 8,57 say "Választás :" get valasz picture "9" range 1,2
     read
     ignem="I"
     if valasz=2
       @ 23,2 say space(77)
       @ 23,10 say "Üzemkész a nyomtató ? (i,n)"
       set console off
       wait to ignem
       set console on
       @ 23,2 say space(77)
     endif
     OK_2=(upper(ignem)="I")
     OK_1=.f.
  else
     @ 23,2 say space(77)
     @ 23,10 say "Ez a személy nem szerepel állományunkban ! Kiván javitani ?
(i,n)"
     set console off
     wait to ignem
     set console on
     @ 23,2 say space(77)
     OK_1=(upper(ignem)="I")
     OK_2=.f.
  endif
enddo

if OK_2
      if valasz=2
         @ 23,2 say space(77)
         @ 23,30 say "Nyomtatás ..."
         set device to print
         set print on
         @ 3,20 say "Adott személy adatainak nyomtatása"
```

```
         @ 4,20 say "---------------------------------"
      endif
      @ 8 ,10 say "Személyi szám            :  " + szezam
      @ 10,10 say "Név                      :  " + v_nev
      @ 12,10 say "Lakcim                   :  " + v_cim
      @ 14,10 say "Iskolai végzettség       :  " + v_iskola+"  "+v_inev
      @ 16,10 say "Beosztás                 :  " + v_beo   +"  "+v_bnev
*
      if valasz=2
         @ 18,5 say replicate("-",70)
         @ 19,5 say " "
         set device to screen
         set print off
      endif
      @ 23,2 say space(77)
      @ 23,20 say "Üssön le egy billentyüt !!"
      set console off
      wait
      set console on
endif
close data
return
```

MODOSIT.PRG

```
*  Személy adatainak módosit :
*  személyi szám bekérése
*  Adatmódositás
*  - név
*  - lakcim
*  - iskolai végzettség (kódolt) -> ISKOLA.DBF  ISKOLA.NDX végzettég kód szerint
*  - beosztás (kódolt)           -> BEOSZTAS.DBF BEOSZTAS.NDX beosztás kód
szerint



clear
@  1,4 to 21,76
@  22,0 to 24,79
@  3,18 say "       Személyi adatok módositása"
@  4,18 say "-------------------------------------"

* adat és indexfile-ok megnyitása

select 1
use szemely index szemely alias SZEM
select 2
use iskola index iskola   alias ISK
select 3
use beosztas index beosztas  alias BEO
sele SZEM

* Az iskola és beosztásmegnevezés
* elökeresése a személykereséssel
```

```
* egyidöben

set relation to iskola into ISK
set relation to beo into BEO

OK_1=.t.
v_szezam=space(13)

do while OK_1
  @ 8,10  say "Személyi szám            : " get v_szezam picture "9-999999-9999"
  read
* utánanézünk hogy az adott személyi számmal
* rendelkezö személy létezik e már adatállományunkban
*
  sele SZEM
  seek v_szezam
  if .not.eof()

    v_nev=nev
    v_cim=cim
    v_iskola=iskola
    v_beo=beo

* a SET RELATION parancs révén
* az ISKOLA file és a BEOSZTAS file mutatója is a SZEMELY.DBF
* ISKOLA és BEO mezöinek megfelelö
* rekordra mutat
*


    @ 14,50 say space(20)
    @ 14,50 say ISK -> inev

    @ 16,50 say space(20)
    @ 16,50 say BEO -> bnev

    OK_1=.f.
    OK_2=.t.
  else
     @ 23,2 say space(77)
     @ 23,10 say "Ez a személy nem szerepel állományunkban ! Kiván javitani ?
(i,n)"
     set console off
     wait to ignem
     set console on
     @ 23,2 say space(77)
     OK_1=(upper(ignem)="I")
     OK_2=.f.
  endif
enddo

do while OK_2

      @ 10,10 say "Név                        : " get v_nev
      @ 12,10 say "Lakcim                     : " get v_cim
      @ 14,10 say "Iskolai végzettség (kód) : " get v_iskola picture "99"
      @ 16,10 say "Beosztás                   : " get v_beo    picture "99"
```

```
*
      read

* Iskolai végzettségkód alapján
* megnevezŐs kikeresése
* és kiirása a képernyöre
*
      sele ISK
      seek v_iskola
      if .not.eof()
        v_inev=inev
      else
        v_inev="Ismeretlen"
      endif

      @ 14,50 say space(20)
      @ 14,50 say v_inev
*
* Beosztáskód alapján
* a megfelelö megnevezŐs kikeresése
* és kiirása a képernyöre
*
      sele BEO
      seek v_beo
      if .not.eof()
        v_bnev=bnev
      else
        v_bnev="Ismeretlen"
      endif

      @ 16,50 say space(20)
      @ 16,50 say v_bnev

      @ 23,2 say space(77)
      @ 23,20 say "Tárolható ez az adat ? (i,n)"
      set console off
      wait to ignem
      set console on

      if upper(ignem)="I"

*  módositott rekord tárolása
*  a SZEMELY.DBF file-ban
*
        sele SZEM
        replace szezam with v_szezam
        replace nev with v_nev
        replace cim with cim
        replace iskola with v_iskola
        replace beo with v_beo
        OK_2=.f.
*
      else
        @ 23,2 say space(77)
        @ 23,20 say "Kiván javitani ? (i,n)"
        set console off
        wait to ignem
```

```
        set console on
        OK_2=(upper(ignem)="I")
      endif
enddo
close data
return
```

TOROL.PRG

```
*  Személy adatainak törlése :
*  személyi szám bekérése
*  rekordtörlés

clear
@  1,4 to 21,76
@  22,0 to 24,79
@  3,18 say "       Személyi adatok törlése"
@  4,18 say "-------------------------------------"

* adat és indexfile-ok megnyitása

select 1
use szemely index szemely alias SZEM
select 2
use iskola index iskola   alias ISK
select 3
use beosztas index beosztas  alias BEO
sele SZEM

* Az iskola és beosztásmegnevezés
* elökeresése a személykereséssel
* egyidöben --> SET REALTION

set relation to iskola into ISK
set relation to beo into BEO

OK_1=.t.
v_szezam=space(13)

do while OK_1
  @ 8,10  say "Személyi szám           : " get v_szezam picture "9-999999-9999"
  read
* utánanézünk hogy az adott személyi számmal
* rendelkezö személy létezik e már adatállományunkban
*
  sele SZEM
  seek v_szezam
  if .not.eof()

    v_nev=nev
    v_cim=cim
    v_iskola=iskola
    v_beo=beo
```

```
* a SET RELATION parancs révén
* az ISKOLA file és a BEOSZTAS file mutatója is a SZEMELY.DBF
* ISKOLA és BEO mezöinek megfelelö
* rekordra mutat
*

    @ 14,50 say space(20)
    @ 14,50 say ISK -> inev

    @ 16,50 say space(20)
    @ 16,50 say BEO -> bnev
    OK_1=.f.
    OK_2=.t.
  else
     @ 23,2 say space(77)
     @ 23,10 say "Ez a személy nem szerepel állományunkban ! Kiván javitani ?
(i,n)"
     set console off
     wait to ignem
     set console on
     @ 23,2 say space(77)
     OK_1=(upper(ignem)="I")
     OK_2=.f.
  endif
enddo

if OK_2
      @ 10,10 say "Név                          : " get v_nev
      @ 12,10 say "Lakcim                       : " get v_cim
      @ 14,10 say "Iskolai végzettség (kód) : " get v_iskola picture "99"
      @ 16,10 say "Beosztás                     : " get v_beo    picture "99"
      clear gets
*
      @ 23,2 say space(77)
      @ 23,20 say "Törölhetö ez az adat ? (i,n)"
      set console off
      wait to ignem
      set console on
      if upper(ignem)="I"
        sele SZEM
* rekord törlés
        delete
        pack
      endif
endif
close data
return
```

12. A hibaüzenetek, rövid magyarázattal

Ez a fejezet a dBASE III PLUS használata közben előforduló hibaüzeneteket és azok értelmezését foglalja össze. A hibaüzeneteket ABC rendben soroltuk fel. A hiba jelentkezésekor a ([])-ben levő hibakódokat is megkaphatjuk.

A DBF file in view is not in current directory

(A környezetet kijelölő file-ban levő DBF file nincs az aktuális könyvtárban.)

A SET VIEW TO vagy a MODIFY VIEW nem találja a környezetet kijelölő file-ban. A file-t vagy töröltük, vagy a PATH SET szerint nem a megfelelő könyvtárban (directory) keressük.

ALIAS name already in use

(Az ALIAS nevet már használjuk)

[24]

Megkíséreltünk egy már megnyitott (USE) adatbázis-file-t használatba venni, amelynek ugyanaz az ALIAS neve, vagy amelynek a neve, vagy az ALIAS-a az A és J tartományon belül van.

ALIAS not found

(Az ALIAS nem található)

[13]

Megkíséreltünk kiválasztani (SELECT) egy adatbázis munkaterületet, de ez kívül esik az A-J tartományon, vagy az ALIAS név nincs definiálva.

ALTERNATE could not be opened

(Az ALTERNATE-t nem lehetett megnyitni)

[72]

A CONFIG.DB file-ban definiált ALTERNATE file nem megnyitható, mert a filenév érvénytelen, vagy a lemezen kevés az üres hely.

A Memo field cannot be selected

(Egy Memo mező nem választható)

Megkíséreltünk kiválasztani vagy bevinni Memo mezőt az ASSIST-ban, amikor az ott nem megfelelő.

At least one file must be in use for this operation

(Legalább egy file-nak használatban kell lennie ennél a műveletnél)

A CREATE/MODIFY VIEW-ban megpróbáltunk hozzáférni az Option menüvel egy adatbázis-file-hoz, anélkül, hogy azt kiválasztottuk volna a Set-Up-ban.

Beginning of file encountered

(File kezdetet talált a rendszer)

[38]

A rekordszámlálót (pointer) a file első rekordja elé pozícionáltuk.

Cannot erase a file which is open

(Nem törölhető nyitott file)

[89]

Megpróbáltunk törölni egy nyitott file-t. Először zárjuk le!

Cannot have subgroups without groups

(Nem nyithatunk alcsoportot csoport nélkül)

A CREATE/MODIFY REPORT-ban megkíséreltünk definiálni egy olyan alcsoportot, amelyhez nem rendelkezünk csoporttal.

Cannot select requested database

(Nem választható a kért adatbázis)

[17]

A kiválasztott (SELECT) munkaterület kívül esik az 1-10 tartományon.

Cannot write to read only file

(Nem lehet írni egy csak olvasható file-ba.)

[111]

Megkíséreltünk egy olyan file-ba írni, amelynek a read/only attributuma be van állítva (SET). Ezt az operációs rendszeren keresztül (DOS funkció segítségével) lehet beállítani.

Command not recognized!

(A parancs nem ismert!)

A megadott parancs nem létezik! (Lehet, hogy csak elgépeltük.)

CONTINUE without LOCATE

(CONTINUE LOCATE nélkül)

Megkíséreltünk végrehajtani egy CONTINUE parancsot, anélkül, hogy előtte egy LOCATE parancsot hajtottunk volna végre.

Cyclic relation

(Ciklikus kapcsolat)

[44]

Végtelen ciklust állítottunk elő a RELATION INTO paranccsal egy olyan adatbázisnál, amely visszahivatkozik az aktuálisan kiválasztott adatbázisra.

Database is encrypted

(Az adatbázis titkosítva van)

[131]

Megkíséreltünk az egyfelhasználós dBASE III PLUS-ban használni egy olyan adatbázist, amely titkosítva volt többfelhasználós verzióban, vagy többfelhasználós dBASE III PLUS-ban megkíséreltünk használni egy titkosított adatbázisfile-t a megfelelő jogosultság nélkül.

Database is not indexed

(Az adatbázis nincs indexelve)

[26]

Az indexfile megnyitása nélkül próbáltunk meg végrehajtani egy FIND, SEEK vagy UPDATE parancsot az adatbázisfile-ban, vagy megkíséreltünk SET RELATION parancsot kiadni arra az adatbázisfile-ra, amely nincs indexelve.

Data Catalog has not been established

(Az adatkatalógus nincs létrehozva)

Akkor kíséreltük meg alkalmazni a lekérdezési záradákot (?) egy parancs részeként, amikor nem volt foglalt katalógus. Ezért használjuk a SET CATALOG TO parancsot a katalógus megnyitására, mielőtt a katalógus lekérdezési záradékot alkalmaznánk a parancsban.

Data type mismatch

(Adattípus összeférhetetlenség)

[9]

Megpróbáltunk egy olyan műveletet végrehajtani, amely operandusai között adattípus összeférhetetlenség van. Nagyon sok feltétel okozhatja ezt a hibaüzenetet. Főként az EXPRESSIONS, REPLACE, SEEK és SORT parancsoknál fordul elő, a következő okokból:

EXPRESSIONS

Azzal próbálkoztunk, hogy nem azonos típusú adatokból állítsunk össze kifejezést.

REPLACE

Eltérő típusú adattal próbáltuk helyettesíteni egy mező tartalmát.

SEEK

A SEEK parancsot olyan kifejezéssel együtt használtuk, amely az aktív indexkulccsal nem egyező adattípusként kerül kiértékelésre.

SORT

Logikai vagy memo mezőn kíséreltünk meg rendezést (SORT).

.DBT file cannot be opened

(A .DBT file nincs megnyitva)

[41]

Megpróbáltunk egy memo típusú mezőt tartalmazó adatbázisfile-t használni (USE), de a dBASE III PLUS nem találja a memo mező szövegét tartalmazó .DBT file-t.

Disk full when writing file: <filename>

(A lemez betelt, amikor a <file-név> nevű file-t írtuk rá)

[56]

A <filename> nevű file már nem fért a lemezre íráskor, így ez nem is történt meg.

DOs nested too deep

(Túl mélyen egymásba skatulyázott DO-k)

[103]

A parancsfile-ok egymásbaágyazásának maximuma 20!

Empty structure will not be saved

(Az üres struktúrát nem mentjük el)

Figyelmeztető üzenet. A CREATE vagy MODIFY STRUCTURE módból olyankor léptünk ki, amikor a struktúra nem tartalmazott mezőt, és a rendszer nem is menti ki.

End-of-file encountered

(File véget talál a rendszer)

[4]

Azzal próbálkoztunk, hogy a rekordmutatót (pointer) az adatbázis utolsó rekordja után pozicionáljuk. Az üzenet megjelenésének négy fő oka lehet. Az alábbiakban ismertetjük azokat a legfontosabb eseteket, amikor előfordulhat ez a hiba.

APPEND FROM, UPDATE FROM, USE

Egy adatbázist nem teljes struktúrával nyitottunk meg.

FIND, SEEK

FIND vagy SEEK parancsot hajtottunk végre sérült vagy nem teljes indexen.

MODIFY LABEL/REPORT, LABEL/REPORT FORM

Olyan LABEL/REPORT FORM file-t nyitottunk meg, amelynek nem teljes a struktúrája, vagy az adatbázisfile struktúrája sérült.

SKIP

A SKIP paranccsal átugrottuk az adatbázisfile utolsó adatrekordját.

End-of-file or error on keyboard input

(File vége vagy hiba a billentyűzetről jövő input-ban)

[51]

A file, amelyet a DOS Set Environment parancsot felhasználva, a billentyűzetről jövő input szimulálására használtunk, megsérült.

Error in HELP system

(Hiba a HELP rendszerben)

Ez a válasz, ha hibás az Assist.hlp file, miközben az Assitant-ban voltunk. Ne használjuk mindaddig, amíg át nem másoljuk a dBASE III PLUS master lemezről a jó változatot.

Exceeded report print width

(Túllépte a jelentés a nyomtatási szélességet)

Azt a szélességet, amelyet a Columns pull-down menüben állítottunk be, a jelentés nyomtatási szélességigénye túllépte, és nem is nyomtatódik ki.

Excess header lines lost

(A többlet fejléc sorok elvesztek)

A CREATE/MODIFY REPORT paranccsal egy olyan hosszú riportformátum fejlécet kíséreltünk meg bevinni, amely a pontosvesszőkkel elválasztva is több, mint négy sor.

Exclusive use on database is required

(Adatbázis kizárólagos használata szükséges)

[110]

Egy olyan dBASE parancsot kíséreltünk meg végrehajtani, amely az adatbázis-file kizárólagos használatát igényli. A SET EXCLUSIVE ON és USE <adatbázis-file> parancsok kiadása után próbálkozzunk újra.

*****Execution error on +: Concatenated string too large**

(Végrehajtási hiba a " + "-nál: Az összefűzendő sztring túl nagy méretű lenne)

[77]

Egy olyan sztringet próbáltunk összefűzni (konkatenálni), aminek az eredménye hosszabb lenne, mint 254 karakter.

*****Execution error on -: Concatenated string too large**

(Végrehajtási hiba a "-"-nál: Az összefűzendő sztring túl nagy méretű lenne)

[76]

Egy olyan sztringet próbáltunk meg összefűzni (konkatenálni), aminek az eredménye hosszabb lenne, mint 254 karakter.

*****Execution error on ^ or ** : Negative base, fractional exponent**

(Végrehajtási hiba a " ^ " vagy "**"-nál: Negatív alap, tört kitevő)

[78]

Egy negatív számot kíséreltünk meg hatványozni, vagy hatványozási kísérlet 0 és 1 közötti hatványra.

*****Execution error on CHR(): Out of range**

(Végrehajtási hiba a CHR()-ben: A megengedett tartományon kívül esik)

[57]

0-nál kisebb, vagy 255-nél nagyobb argumentumot használtunk a CHR() függvényben.

*****Execution error on LOG(): Zero or negative**

(Végrehajtási hiba a LOG()-ban: Nulla vagy negatív)

[58]

A LOG() függvényben 0-t vagy negatív számot használtunk argumentumként.

*****Execution error on NDX(): Invalid index number**

(Végrehajtási hiba az NDX()-ben: Érvénytelen indexszám)

[87]

Az NDX() függvény argumentumaként egynél kisebb vagy hétnél nagyobb számot használtunk.

*****Execution error on REPLICATE(): String too large**

(Végrehajtási hiba a REPLICATE()-ben: A sztring túl hosszú)

[88]

Azzal próbálkoztunk, hogy egy 254 karakternél hosszabb sztringet állítsunk elő.

*****Execution error on SPACE(): Negative**

(Végrehajtási hiba a SPACE()-nél: Negatív)

[60]

Negatív argumentummal használtuk a SPACE() függvényt.

*****Execution error on SPACE(): too large**

(Végrehajtási hiba a SPACE()-nél: túl nagy)

[59]

A SPACE() függvényt olyan argumentummal használtuk, amely 254-nél nagyobb.

*****Execution error on SQRT(): Negative**

(Végrehajtási hiba a SQRT()-nél: Negatív)

[61]

Négyzetgyök függvény argumentumaként negatív számot használtunk.

*****Execution error on STORE: String too large**

(Végrehajtási hiba a STORE-nál: túl hosszú sztring)

[79]

A maximálisan megengedett 254 karakternél nagyobb méretű memóriaváltozót kíséreltünk meg előállítani.

*****Execution error on STR(): Out of range**

(Végrehajtási hiba a STR()-nél: tartományon kívül esik)

[63]

A STR() függvényt nem megengedett vagy összeegyeztethetetlen paraméterekkel használtuk. A hosszúság nem lehet több 19-nél, s a tizedes jegyek száma nem haladhatja meg a hosszúság minusz kettő értéket.

*****Execution error on STUFF(): String too large**

(Végrehajtási hiba a STUFF()-nál: túl hosszú sztring)

[102]

A STUFF() függvény eredményeként a maximálisan megengedett 254 karakternél hosszabb sztringet kaptunk volna.

*****Execution error on SUBSTR(): Start point out of range**

(Végrehajtási hiba a SUBSTR()-nél: kiindulási pont a megengedett tartományon kívül esik)

[62]

A részsztring kezdőpozíciójának olyan értéket adtunk meg, amely eleve nagyobb, mint annak a sztringnek a hossza, amiből ki kellene emelni.

^ ^ Expected ON or OFF

(Csak ON vagy OFF elfogadható)

[73]

A CONFIG.DB file-ban szereplő paraméterét ON vagy OFF helyett másra állítottuk be.

Field name is already in use

(A mezőnév már használatban van)

Az adatbázis struktúrájának beállítása vagy módosítása (CREATE/MODIFY STRUCTURE) közben egynél többször próbáltuk meg felhasználni ugyanazt a mezőnevet.

Field not found

(A mező nem található)

[48]

A BROWSE-ban olyan mezőt próbáltunk meg rögzíteni (FREEZE), amelyik nem is létezik.

Field type must be C,N,D,L or M

(A mező típusának C,N,D,L vagy M-nek kell lennie)

A struktúra létrehozásban/módosításban (CREATE/MODIFY STRUCTURE) karakteres, numerikus, dátum, logikai vagy memo típusútól eltérő típusú mezőt kíséreltünk meg generálni.

File already exist

(A file már létezik)

[7]

Megpróbáltunk átnevezni egy file-t olyan névre, amely már létezik a lemezen.

File does not exist

(A file nem létezik)

[1]

Egy nemlétező file-t próbáltunk megnyitni.

File is already open

(A file már meg van nyitva)

[3]

Egy megnyitott file-t próbáltunk újból megnyitni.

File is in use by another

(A file-t más használja)

[108]

Egy olyan file-t próbáltunk megnyitni, amelyet kizárólagos megnyitási móddal (EXCLUSIVE) már valaki más használ. Ez lehet olyan adatfile, amelyet a SET EXCLUSIVE kapcsoló ON állásánál nyitottunk meg, vagy lehet olyan file, amelyhez a file kizárólagos használatát megkövetelő paranccsal nyúlunk.

File is not accessible

(A file nem elérhető)

[29]

Megpróbáltunk egy lemezes file-t létrehozni akkor, amikor a lemez megtelt, vagy a file neve egy vagy több érvénytelen karaktert tartalmazott (+"< >?*[]).

File too large, some data may be lost

(A file túl nagyméretű, lehet, hogy néhány adat elveszett)

Megkíséreltünk egy nagyméretű szövegfile-t, vagy memo mezőt javítani a dBASE III PLUS saját szövegszerkesztőjével. A szövegszerkesztővel kezelhető legnagyobb méretű file 5000 karakteres lehet.

File was not LOADed

(A file nincs betöltve a memóriába)

[91]

Egy olyan file-t kíséreltünk meg betölteni a memóriába, amely nem tölthető be, mert nulla hosszúságú, vagy nem áll rendelkezésre, illetve azzal próbálkoztunk, hogy egy olyan modult hívjunk meg (CALL), vagy szabadítsunk fel (RELEASE), amelyik nem töltődött be.

HELP text not found

(A HELP szövege nem található)

A pont promptból kértük a HELP-et, és a HELP.DBS file nincs ugyanazon a lemezen vagy könyvtárban, mint ahol a DBASE.OVL.

Illegal character data length

(Illegális karakter adathossz)

A CREATE FROM-ban megadott adatfile-ban a mezőspecifikáció hossza nincs az 1-től 254-ig terjedő tartományban. (Ez, ha azt a STRUCTURE EXTENDED segítségével hoztuk létre, lehetetlen, de maga az adatfile átírható.)

Illegal data length

(Illegális adathosszúság)

Karakter típusú mezőt definiáltunk CREATE/MODIFY STRUCTURE módban, de annak hossza nincs az 1-től 254-ig terjedő tartományban.

Illegal decimal length

(A tizedes jegyek száma nem megfelelő)

Meghatároztunk egy numerikus típusú mezőt, de a tizedes helyek száma kisebb, mint 0, vagy nagyobb mint 15, vagy nagyobb mint a teljes numerikus mező szélessége minusz kettő.

Illegal field name

(Illegális mezőnév)

Megadtunk egy struktúra kiterjesztésű file-t a CREATE FROM-ban, amelyben illegális karakter(eke)t tartalmazó mezőnév specifikáció van. (Ez, ha azt a STRUCTURE EXTENDED segítségével hoztuk létre, lehetetlen, de maga az adatfile átírható.)

Illegal field type

(Illegális mezőtípus)

Megadtunk egy struktúra kiterjesztésű file-t a CREATE FROM-ban, amelyben illegális karakter(eke)t tartalmazó mezőnév specifikáció van. (Ez, ha azt a STRUCTURE EXTENDED segítségével hoztuk létre, lehetetlen, de maga az adatfile átírható.)

Illegal numeric data length

(Illegális numerikus adathosszúság)

Egy olyan numerikus típusú mezőt határoztunk meg, amely hosszúsága kisebb mint 1, vagy nagyobb mint 19.

Illegal value

(Illegális érték)

[46]

A teljes képernyős SET parancs végrehajtása közben az Esc billentyűt nyomtuk le olyankor, amikor be kellett volna állítani a tizedeseket vagy a margót.

Improper data type in subsubtotal expression

(Nem megfelelő adattípus a rész-részösszeg kifejezésben)

Olyan REPORT FORM programot futtattunk, amelynek a logikai vagy memo adat típusa a rész-részösszegben van meghatározva.

Improper data type in subtotal expression

(Nem megfelelő adattípus a részösszeg kifejezésben)

Olyan REPORT FORM programot futtattunk, amelynek a logikai vagy memo adat típusa a részösszegben van meghatározva.

Index damaged. REINDEX should be done before using data

(Az index megsérült. Indexeljük újra, mielőtt az adatokat használjuk.

[114]

Olyan indexfile-t próbáltunk használni, amellyel való indexelés közben Esc-et adtunk, s amely megszakította az indexelést. Elképzelhető, hogy az indexfile fizikailag sérült.

Index expression is too big (220 char maximum)

(Az indexkifejezés túl nagy (220 karakter a maximum))

[112]

Az az indexkifejezés, amellyel indexeltünk, hosszabb, mint 220 karakter. Ez a határérték a kulcskifejezés karaktereinek számára vonatkozik, és nem a kulcskifejezés értékeként kapott karakterlánc hosszára, ami csak 100 lehet.

Index file does not match database

(Az indexfile nem felel meg az adatbázisnak)

[19]

Olyan indexfile megnyitásával próbálkoztunk, amelynek kulcskifejezése nem kapcsolódik a használatban lévő adatbázishoz.

Index interrupted. Index will be damaged if not completed.

(Indexelés megszakítása. Az index sérült lesz, ha nem fejezzük be)

[113]

Lenyomtuk az Esc billentyűt, mielőtt az INDEX ON vagy a REINDEX parancs befejeződött volna. Ez a hibaüzenet lehetőséget ad arra, hogy folytassuk vagy abbahagyjuk az indexelést.

Index is too big (100 char maximum)

(Az index túl nagy (100 karakter a maximum))

[23]

Az indexelés kulcskifejezésének értéke meghaladja a 100 karaktert. Ez a határérték a kulcskifejezés tartalmának kiértékelésére vonatkozik, nem pedig magának a kifejezésnek a hosszára.

Insert impossible

(Beszúrás lehetetlen)

A CREATE/MODIFY REPORT-ban megpróbáltunk beszúrni még egy mezőt akkor, amikor ott már 24 mező volt.

Insufficient dynamic memory for contents window

(Nincs elegendő dinamikus memória a contents "ablaka" számára)

Számítógépünknek nincs elegendő szabad memóriája arra, hogy bemutassa a Contents pull-down menü tartalmát a CREATE/MODIFY LABEL-ben.

Insufficient memory

(Nincs elegendő memória)

[43]

Megkíséreltük betölteni a dBASE III PLUS-t, de a számítógép RAM memóriája kisebb, mint 256 kbyte. Emellett számos más feltétel is okozhatja ennek a hibaüzenetnek a megjelenését:

@...GET

Túlléptük a GET-ek maximális számát (128) anélkül, hogy kiadtunk volna egy CLEAR GETS vagy egy CLEAR parancsot.

RUN, TEDIT, WP

Megpróbáltunk futtatni (RUN) egy külső programot elégtelen memóriaméret mellett. A szükséges memóriaméret: MAXMEM + a külső program mérete −17 kbyte még a COMMAND.COM számára. A MAXMEM értéke 256 kbyte, kivéve, ha módosítottuk a CONFIG.DB-ben.

LOAD

Megpróbáltunk betölteni egy olyan file-t, amely nagyobb, mint a rendelkezésre álló memória mérete.

Insufficient memory to display fields

(Nincs elegendő memória a mezők megjelenítésére)

Számítógépünk nem rendelkezik elegendő szabad memóriával ahhoz, hogy megjelenítsük a kért mezőt a pull-down menüben.

Internal error: CMDSET():

(Belső hiba: CMDSET():)

[66]

Probléma van a dBASE III PLUS rendszerrel. Készítsünk egy új munkamásolatot a dBASE III PLUS master lemezről. Ha ez a hiba még mindig fennáll, akkor vegyük fel a kapcsolatot az Ashton-Tate Software Support-tal.

Internal error: EVAL work area overflow

(Belső hiba: EVAL munkaterület túlcsordulása)

[67]

Egy olyan kifejezést kíséreltünk meg használni, amely túlságosan összetett és ezért nem értékelhető ki. Bontsuk több részre a kifejezést, és próbálkozzuk újra.

Internal error: Illegal opcode

(Belső hiba: Illegális műveleti kód)

[68]

Probléma van a dBASE III PLUS rendszerrel. Készítsünk egy új munkamásolatot a dBASE III PLUS master lemezről. Ha még mindig ezt a hibát kapjuk, akkor vegyük fel a kapcsolatot az Ashton-Tate Software Support-tal.

Ez a hiba előfordulhat egy kb. 150-200 utasításszóból álló összetett kifejezés végrehajtása alatt is.

Internal error: Unknown command code

(Belső hiba: Ismeretlen parancskód)

[65]

Probléma van a dBASE III PLUS rendszerrel. Készítsünk egy új munkamásolatot a dBASE III PLUS master lemezről. Ha még mindig ezt a hibát kapjuk, akkor vegyük fel a kapcsolatot az Ashton-Tate Software Support-tal.

*****INTERRUPTED*****

(Megszakítva)

A parancs vagy parancsfile végrehajtását az Esc billentyű lenyomásával megszakítottuk.

Invalid date (press SPACE)

(Érvénytelen dátum (nyomja le a szóközt))

[81]

Érvénytelen dátumot próbáltunk bevinni dátum típusú mezőbe vagy memóriaváltozóba. A kilépéshez nyomjuk le a szóköz billentyűt.

Invalid DIF Character

(Érvénytelen DIF karakter)

[118]

Érvénytelen DIF file-t kíséreltünk meg egy APPEND FROM műveletben használni. A hiba oka feltehetően a DIF file-ban levő ellenőrző karakter.

Invalid DIF File Header

(Érvénytelen DIF file fejrész)

[115]

Érvénytelen fejrészű DIF file-t kíséreltünk meg egy APPEND FROM műveletben használni.

Invalid DIF Type Indicator

(Érvénytelen DIF típus jelző)

[117]

Érvénytelen adattípus jelzőt tartalmazó DIF file-t kíséreltünk meg egy APPEND FROM műveletben használni.

Invalid DOS SET option

(A DOS SET parancsának érvénytelen opciója)

[99]

A DOS SET parancsát érvénytelen paraméterrel próbáltuk használni.

Invalid function argument

(Érvénytelen függvény argumentum)

[11]

Egy függvényt olyan argumentummal próbáltunk használni, amilyen típusút e függvény nem fogad el.

Invalid function name

(Érvénytelen függvénynév)

[31]

Olyan függvényt adtunk meg, amelyet a dBASE III PLUS nem ismer.

Invalid index number

(Érvénytelen index szám)

[106]

Egy SET ORDER TO paranccsal az indexfile listában olyan helyre próbáltunk pozícionálni, amelyen nincs indexfile, vagy kívül esik a 0-tól 7-ig terjedő tartományon.

Invalid operator

(Érvénytelen művelet)

[107]

Egy dBASE kifejezésben olyan műveletet használtunk, amely a kifejezésben használt adattípusok között érvénytelen.

Invalid printer port

(Érvénytelen nyomtató bemenet (port))

[123]

Megpróbáltuk a nyomtatásra (SET PRINTER TO) kiválasztani az operációs rendszer egy olyan eszközét, amely nem létezik. Ellenőrizzük az operációs rendszer kézikönyvében az érvényes nyomtató eszközök listáját.

Invalid printer redirection

(Érvénytelen printer átirányítás)

[124]

A SET PRINTER TO paranccsal a hálózati nyomtatót próbáltuk meg kiválasztani, de a megadott nyomtató nem létezik.

Invalid SYLK File Dimension Bounds

(Érvénytelen SYLK file dimenzió határok)

[120]

Megkíséreltünk egy APPEND FROM műveletet végrehajtani egy SYLK file-on, amely egy vagy több, a file megadott határain kívülre mutató adatot tartalmaz.

Invalid SYLK File Format

(Érvénytelen SYLK file formátum)

[121]

Megkíséreltünk APPEND FROM műveletet végrehajtani egy SYLK file-on, amely vagy nem MultiPlan-nal készült, vagy valamiképp megsérült.

Invalid SYLK File Header

(Érvénytelen SYLK file fejrész)

[119]

Megkíséreltünk APPEND FROM műveletet végrehajtani egy SYLK file-on, amelynek érvénytelen fejrésze van.

^ - Keyword not found

(A ^ jellel határolt kulcsszó nem található)

[86]

A CONFIG.DB-ben azonosíthatatlan kulcsszó van.

Label contents cannot be Memo type

(A cimke tartalma nem lehet memo típusú)

A CREATE/MODIFY LABEL opcióval cimke tartalomnak memo típusú mezőt próbáltunk adni. Csak karakter, numerikus, dátum és logikai típusok megengedettek.

Label file invalid

(Érvénytelen cimke-file)

[54]

A .LBL file formája nem megfelelő, elképzelhető, hogy a file megsérült vagy nem a CREATE/MODIFY LABEL paranccsal hoztuk létre.

Line exceeds maximum of 254 characters

(A sor meghaladta a 254 karakteres maximumot)

[18]

Olyan parancsot akartunk használni parancs (.prg) vagy formátum (.fmt) fileban, amely meghaladja a 254 karakteres maximális hosszúságot.

Max number of fields already reached

(Már elértük a mezők maximális számát)

A CREATE vagy a MODIFY STRUCTURE parancsban megpróbáltunk hozzáadni még egy mezőt az adatbázis-file-hoz akkor, amikor annak már 128 mezője volt.

Maximum record length exceeded

(Túlléptük a maximális rekordhosszt)

[137]

A CREATE vagy a MODIFY STRUCTURE parancsban egy olyan mezőt kíséreltünk meg definiálni, amellyel a rekord teljes hossza már meghaladta volna a 4000 byte-ot. Másik lehetséges ok, hogy a CREATE FROM-ban a mezők méreteinek összege meghaladta a 4000 byte-ot.

Memory Variable file is invalid

(A memóriaváltozó file érvénytelen)

[55]

Olyan file-t adtunk meg a RESTORE paramétereként, amely formátuma nem megfelelő. Elképzelhető, hogy elírtuk a file nevét, vagy a file megsérült.

Mismatched DO WHILE and ENDDO

(Nem egymáshoz tartozó DO WHILE és ENDDO)

[96]

A dBASE III PLUS egy felesleges ENDDO-val találkozott a parancsfile-ban.

No database is in USE. Enter file name:

(Nincs használatban levő adatbázis-file. Adjuk meg a file nevét:)

[52]

Olyan parancsot specifikáltunk, amely egy adatbázis-file megnyitását igényli az aktuális munkaterületen. Írjuk be a használni kívánt adatfile nevét, és nyomjuk meg az Enter billentyűt. Ennek hatására a rendszer végrehajt egy pótlólagos USE parancsot.

No fields of the required type are present

(Nincs jelen a szükséges típusú mezőből egy sem)

A dBASE III PLUS menüvezérelt parancsainak egyikében kértünk egy fields pull-down menüt (kiválasztással vagy az F10 billentyű lenyomásával), és nem volt a szükséges típusú mező sem a file struktúrában, sem a SET FIELDS listájában.

No fields to process

(Nincs feldolgozandó mező)

[47]

Az ASSIST segítségével megpróbáltunk összeadni (SUM) vagy számtani átlagot (AVERAGE) számítani egy adatbázis file-ra, de annak nem volt numerikus mezője.

No find

(Nincs találat)

[14]

Megkíséreltünk végrehajtani egy FIND vagy SEEK parancsot egy kulcskifejezéssel, de az nem egyezett a megadott indexfile-ban levő egyetlen bejegyzéssel sem.

No PARAMETER statement found

(Nem található PARAMETER utasítás)

[93]

Megpróbáltunk végrehajtani egy paraméteres parancsfile-t vagy egy eljárást (DO...WITH...) akkor, amikor nem volt PARAMETER parancs a program vagy eljárás file első soraként megadva.

No room for heading

(Nincs hely a fejrész számára)

A fejrész túllépte a CREATE/MODIFY REPORT-ban a meghatározott méretet.

Not a character expression

(Nem karakter kifejezés)

[45]

Ott próbáltunk meg karakter típusú kifejezést használni, ahol ez nem megengedett.

Not a dBASE database

(Nem dBASE adatbázis)

[15]

Olyan file-t próbáltunk használni, amely nem felel meg a dBASE formátumának. Lehet, hogy sérült a fejrész rekordja, vagy elírtuk a file nevét.

Not a Logical expression

(Nem logikai kifejezés)

[37]

Nem logikai típusú kifejezést használtunk ott, ahol logikai kifejezés szükséges; mint például a FOR/WHILE feltételben.

Not a numeric expression

(Nem numerikus kifejezés)

[27]

Nem numerikus típusú kifejezést használtunk ott, ahol numerikus kifejezés szükséges, mint például a SUM, AVERAGE, DISPLAY NEXT parancsokban.

Not a valid QUERY file

(Nem érvényes QUERY file)

[134]

A SET FILTER TO parancsot érvénytelen QUERY file-ra próbáltuk kiadni. A QUERY file már valószínüleg sérült, és azt a CREATE QUERY paranccsal újra elő kell állítani.

Not a valid VIEW file

(Nem érvényes VIEW file)

[127]

SET VIEW TO vagy egy MODIFY VIEW parancsot próbáltunk kiadni egy érvénytelen környezetet kijelölő file-ra. A környezetet kijelölő file már valószínüleg sérült, és azt újra elő kell állítani a CREATE VIEW FROM ENVIRONMENT vagy a CREATE VIEW segítségével. Az adatbázis és más file-ok a környezetet kijelölő file-ban még épek.

Not enough disk space for SORT

(Nincs elegendő lemezterület a SORT számára)

Nincs elég hely a lemezen az adatbázis-file sorbarendezésére (SORT). A lemezen a forrás adatbázis-file-nak megfelelő nagyságú két file számára szükséges területnek kell lennie, mert a SORT a célfile-on kívül egy ideiglenes file-t is használ.

Not enough records to sort

(Nincs elég rekord a sorbarendezéshez)

Egy olyan adatbázis-file-t kíséreltünk meg sorbarendezni (SORT), amelynek még két rekordja sincs.

****Not found****

(Nem található)

[82]

Olyan keresési sztringet határoztunk meg, amely a BROWSE FIND menüopciójával nem található.

Not readable

(Nem olvasható)

A DIR parancsot használtuk, de a dBASE III PLUS képtelen volt olvasni a file-t (Valaki már azt zárolta). Nem hibaüzenet.

Not suspended

(Nincs felfüggesztve)

[101]

A RESUME parancsot olyankor adtuk ki, amikor a parancsfile végrehajtása nem volt felfüggesztve.

Numeric overflow (data was lost)

(Numerikus túlcsordulás (az adat elveszett))

[39]

Numerikus túlcsordulás következik be, mert egy matematikai művelet eredménye túl nagy ahhoz, hogy a kijelölt mezőben vagy változóban letároljuk.

Azt a két egyszerű feltételt, amely ezt kiválthatja, az alábbiakban részletezzük:

REPLACE

A numerikus mezőbe a REPLACE...WITH... paranccsal egy túlságosan nagy kifejezést próbáltunk behelyettesíteni.

TOTAL

Az általunk specifikált numerikus mező túl kicsi ahhoz, hogy be tudja fogadni az adatbázis összegzésének eredményét (TOTAL).

Operation with Logical field invalid

(A művelet logikai mezővel érvénytelen)

[90]

Logikai mező rendezésével (SORT) vagy indexelésével (INDEX) próbálkoztunk.

Operation with Memo field invalid

(A művelet memo mezővel érvénytelen)

[34]

Memo mező rendezésével (SORT) vagy indexelésével (INDEX) próbálkoztunk, illetve egy változóhoz memo mezőt akartunk rendelni.

Out of memory variable memory

(A memóriaváltozó túlcsordult a memórián)

[21]

Túlléptük a memóriaváltozók számára lefoglalt 6000 byte-os szabad memória helyet. Ez a korlát kiterjeszthető a CONFIG.DB file-ban az MVARSIZ paraméter megváltoztatásával.

Out of memory variable slots

(A memóriaváltozó túllépett a lehetőségeken)

[22]

Megkíséreltünk egy memóriaváltozót akkor inicializálni, amikor már 256 memóriaváltozó aktív volt.

^ --- Out of range

(^ --- A határokon kívül esik)

[75]

A CONFIG.DB file-ban egy olyan paramétert specifikáltunk, amely nincs a megengedett határokon belül.

Position is off the screen

(A pozíció a képernyőn kívül esik)

[30]

Megpróbáltunk végrehajtani egy olyan SAY/GET parancsot, amelynél az oszlop pozíció a 0-79 határokon, vagy a sor pozíció a 0-24 határokon kívül esik.

Printer is not installed on port

(A printer nincs installálva a kimeneten)

[126]

Olyan kimenetre kíséreltünk meg kiadni egy SET PRINTER TO parancsot, amelyhez nem csatlakozik nyomtató.

Printer not ready

(A printer nem üzemkész)

[125]

Olyankor próbáltunk nyomtatni, amikor a printer nem volt csatlakoztatva, vagy nem volt a hálózati áramkörbe kapcsolva, vagy nem volt bekapcsolva. Lehetőségünk nyílik arra, hogy kiküszöböljük a hibát, és újrapróbáljuk a parancsot, vagy megszakítsuk a parancsot, vagy figyelmen kívül hagyjuk a hibát.

Az említett hiba megszüntetésére az egyetlen biztonságos módszer, ha a nyomtató problémáját megszüntetjük, és újra próbálkozunk.

Record is in use by another

(A rekordot más használja)

[109]

A rekord zárolása nem lehetséges, mert valaki más már zárolta. Valójában nem hibaüzenet, tovább kell kísérletezni a rekord zárásával, míg nem sikerül.

Record is not in index

(A rekord nincs az indexben)

[20]

A rekordpointert a GO/GOTO segítségével vagy a RECORD paranccsal megpróbáltuk egy olyan rekordhoz mozgatni, amely nincs az indexfile-ban.

Record is not locked

(A rekord nincs zárolva)

[130]

Olyan rekordot próbáltunk módosítani többfelhasználós üzemmódban, amelyiket előzőleg nem zároltunk. Próbáljuk meg zárolni, s azután módosítani.

Record is out of range

(A rekord a határokon kívül esik)

[5]

Olyan rekordszámra próbáltuk beállítani a rekordmutatót, amely nagyobb, mint az adatbázis file-ban levő rekordok száma.

Record not inserted

(A rekord nincs beszúrva)

[25]

Befejeztünk egy INSERT parancsot anélkül, hogy a rekord hozzáfűződött volna az adatbázis-file-hoz.

Records do not balance (program error)

(A rekordok nincsenek egyensúlyban (programhiba))

Valamilyen probléma van a dBASE III PLUS rendszerrel. Készítsünk egy új munkamásolatot a dBASE III PLUS master lemezről. Ha még mindig hibát kapunk, vegyük fel a kapcsolatot az Ashton-Tate Software Support-tal.

Report file invalid

(Érvénytelen jelentés file)

[50]

Olyan REPORT FORM (.frm) file futtatásával próbálkoztunk, amelyet nem a CREATE vagy MODIFY REPORT-tal állítottunk elő.

Structure invalid

(Érvénytelen struktúra)

[33]

Akkor próbálkoztunk CREATE FROM parancs végrehajtásával, amikor az a struktúra, amelyet a struktúra-kiterjesztés file-ban definiáltunk, nem érvényes.

Syntax error

(Szintaktikai hiba)

[10]

A végrehajtandó parancs szintaktikai (formai) hibát tartalmaz.

Syntax error in contents expression

(Szintaktikai hiba tartalmi kifejezésben)

Hibát vétettünk egy LABEL mező tartalmában.

Syntax error in field expression

(Szintaktikai hiba mező kifejezésben)

Szintakikai hibát vétettünk egy REPORT mező tartalmában.

Syntax error in group expression

(Szintaktikai hiba csoport kifejezésben)

Szintaktikai hibát vétettünk egy REPORT csoport kifejezésében.

Syntax error in subgroup expression

(Szintaktikai hiba egy alcsoport kifejezésben)

Szintaktikai hibát vétettünk egy REPORT alcsoport kifejezésében.

Table is full

(A tábla tele van)

[105]

A maximális öt file-nál több betöltésével (LOAD) kísérleteztünk.

The HELP file is not available

(A HELP file nem áll rendelkezésre)

HELP-et kértük az ASSIST-ban, és az ASSIST.HLP file nincs azonos lemezen vagy könyvtárban a DBASE.OVL-lel.

There are no files of the type requested on this drive

(Ezen a meghajtón nincsenek meg a kért típusú file-ok)

[53]

Az ASSIST-ban és a SET-ben vagy más menüvezérelt parancsokban egy olyan menüopciót választottunk ki, amely az összes megadott típusú file-t tartalmazó almenüt kéri, és a szükséges file-ok nincsenek az alapértelmezés szerinti meghajtón és könyvtárban.

This field must contain a valid dBASE expression

(Ennek a mezőnek érvényes dBASE kifejezést kell tartalmaznia)

A CREATE/MODIFY QUERY-ben egy érvénytelen kifejezést adtunk meg a Constant/Expression opció részére a Set Filter pull-down menüjében.

A CREATE/MODIFY REPORT-ban egy érvénytelen kifejezést vittünk be a Groups vagy a Columns pull-down menübe.

Mindkét esetben elegendő, ha ellenőrizzük a mező nevét betűnként, és a kifejezésben használt műveletek érvényességét.

This must be a valid dBASE expression

(Ennek érvényes dBASE kifejezésnek kell lennie)

Megpróbáltunk egy érvénytelen kifejezést beírni az aktuális cimke tartalomsorba a CREATE/MODIFY LABEL alatt. A Label tartalom vagy egy karaktert tartalmazó karakteres kifejezés, vagy ezek listája.

This string must not contain any semicolons

(Ez a sztring nem tartalmazhat pontosvesszőt)

A CREATE/MODIFY REPORT-ban egy vagy több pontosvesszőt kíséreltünk meg beírni az Options pull-down menüben a Title opcióba, vagy a Columns pull-down menüben a Heading opcióba. Ezekben az esetekben a pontosvessző használata nem megengedett. Ahhoz, hogy egynél több Title vagy oszlop Heading sor legyen, használjuk az ezen opciókra megadott és a beviteli területen biztosított négy sort.

Too many characters in REPORT

(Túl sok karakter van a REPORT-ban)

A REPORT FORM paramétereket az 1440 byte-os felső korlátot túllépve határoztuk meg.

Too many files are open

(Túl sok file van megnyitva)

[6]

Megkíséreltünk 15 file-nál többet megnyitni. Ezt az üzenetet kapjuk akkor is, ha a DOS konfigurációs file-ja (CONFIG.SYS), amelynek tartalmaznia kell az egyszerre megnyitható file-ok számát (FILES = 20), nincs a boot lemez gyökér könyvtárában.

Too many indices

(Túl sok index)

[28]

Egy USE vagy SET INDEX TO parancs mellett hétnél több index-file meghatározásával próbálkoztunk.

Too many merge steps

(Túl sok összeválogatási lépés)

Probléma a dBASE III PLUS rendszerrel. Készítsünk egy új munkamásolatot a dBASE III PLUS master lemezről. Ha még mindig fennáll a hiba, vegyük fel a kapcsolatot az Ashton-Tate Software Support-tal.

Too many sort key fields

(Túl sok rendezési kulcsmező)

Tíznél több mezőre próbáltunk SORT ON parancsot kiadni.

Total Label width exceeds maximum allowable size

(A teljes cimkeszélesség meghaladja a megengedett maximumot)

Olyan LABEL FORM-ot próbáltunk futtatni, ahol a cimkeszélesség nagyobb, mint 250 karakter.

^ --- Truncated

(Csonkítva)

[74]

A CONFIG.DB-ben olyan parancssort próbáltunk létrehozni, amely túl hosszú és ezért csonkítva lett.

Two files must be in use in order to set a relation

(Két file-t kell használni egy reláció beállítására)

A CREATE/MODIFY VIEW-ben olyankor próbálkoztunk a Relate pull-down menüvel, amikor csak egy adatbázis file-t választottunk ki a Set-up-ban.

Unable to load COMMAND.COM

(Képtelen a COMMAND.COM-ot betölteni)

[92]

A COMMAND.COM program nincs az "A" meghajtón, vagy a DOS SET COMSPEC parancsban specifikált meghajtón akkor, amikor:

- kiadtunk egy RUN/! parancsot
- végrehajtottuk a MODIFY COMMAND-ot, vagy bevittük a memo mezőt, amikor egy külső szövegszerkesztő volt kijelölve a CONFIG.DB-ben a TEDIT-hez vagy a WP-hez.

Unable to lock

(Képtelen zárolni)

[129]

Teljes képernyős szerkesztési üzemmódban a <Ctrl-O> lenyomásával megkíséreltük zárolni a rekordot, de nem sikerült, mert valaki más már zárolta. Ez a jelzés csak többfelhasználós üzemmód esetén jöhet elő.

Unable to Skip

(Képtelen ugrani)

[128]

Olyan rekordra akartunk ugrani (SKIP), amit egy másik felhasználó zárolt, így annak tartalma nem olvasható. Csak többfelhasználós üzemmód esetén jöhet elő.

Unassigned file no.

(Hozzárendeletlen file-szám)

[2]

Probléma van a dBASE III PLUS rendszerrel. Készítsünk új munkamásolatot a dBASE III PLUS master lemezéről. Ha még mindig fennáll a hiba, vegyük fel a kapcsolatot az Ashton-Tate Software Support-tal.

Unauthorized access level

(Jogosulatlan hozzáférési szint)

[133]

Megpróbáltunk használni egy mezőt vagy egy file-t, amelyhez a miénkkel nem egyező elérési szint tartozik.

Unauthorized login

(Jogosulatlan bejelentkezés)

[132]

Háromszor egymás után sikertelenül jelentkeztünk be a dBASE programba.

Unbalanced parenthesis

(A zárójelek páratlanok)

[8]

Egy kifejezésben nem találkoznak a nyitó- és a záró zárójelek. A CREATE/ MODIFY QUERY-ben a Nest pull-down menüvel hozzáadtunk vagy eltávolítottunk egy zárójelet, és elmulasztottuk hozzáadni vagy eltávolítani a párját.

Unknown function key

(Ismeretlen funkcióbillentyű)

[104]

A SET FUNCTION-ban egy olyan funkcióbillentyűt specifikáltunk, amely nem egyezik meg egyik létező funkcióbillentyű nevével vagy számával sem.

Unknown SCEDIT() return code:

(Ismeretlen SCEDIT() visszatérési kód:)

Probléma van a dBASE III PLUS rendszerrel. Készítsünk egy új munkamásolatot a dBASE III PLUS master lemezről. Ha még mindig hibajelzést kapunk, vegyük fel a kapcsolatot az Ashton-Tate Software Support-tal.

*****Unrecognized command verb**

(Azonosíthatatlan parancs)

[16]

A megadott parancs nem létezik.

Unrecognized phrase/keyword in command

(Azonosíthatatlan kifejezés/kulcsszó a parancsban)

[36]

Egy kifejezés kezdetén érvénytelen kulcsszó áll.

Unsupported path given

(Nem támogatott elérési útvonal megadása)

[136]

Megpróbáltunk előállítani (CREATE) egy adatbázis-file-t, és elérési útvonalként a "..\"-t használtuk.

Unterminated string

(Lezáratlan sztring)

[35]

Egy sztring kifejezést lezáró határolójel nélkül specifikáltunk. Az érvényes határoló karakterek: az egyszeres idézőjel, a dupla idézőjel és a szögletes zárójelpárok.

Valid only in programs

(Csak programon belül érvényes)

[95]

Ezen parancsok valamelyikét adtuk ki a pont promptról: CASE, DO CASE, DO WHILE, ELSE, ENDCASE, ENDDO, ENDIF, ENDTEXT, IF, LOOP, OTHERWISE, SUSPEND, TEXT.

Variable not found

(A változó nem található)

[12]

Egy olyan nem létező memóriaváltozót vagy változónevet specifikáltunk, amely az aktuálisan kiválasztott munkaterületen, vagy álnévvel specifikált munkaterületen egy nem létező mező.

****WARNING** Data will probably be lost. Confirm (Y/N)**

(**FIGYELMEZTETÉS** Az adat valószínüleg el fog veszni. Rendben van-e így (Igen/Nem))

[70]

Az abort opciót választottuk a "disk full" (a lemez megtelt) üzenetnél.

****WARNING** Report form empty**

(**FIGYELMEZTETÉS** A jelentés formátum üres)

CREATE REPORT-ból való kilépéssel próbálkoztunk akkor, amikor még a jelentés számára nem volt mező definiálva.

Wrong number of parameters

(Hibás számú paraméter)

[94]

Nem egyező számú paraméterekkel pontatlanul állítottuk be a DO WITH és a PARAMETERS parancsokat.

dBCODE végzetes és súlyos hibák

A végzetes hibák a dBCODE abortálását okozzák. A súlyos hibák esetén a dBCODE törli a kimenő file-t, de a feldolgozás folytatódik.

Az alábbiakban ismertetjük a dBASE III PLUS Run Time + végzetes és súlyos hibaüzeneteinek listáját.

Végzetes hibák

*****Ran out of memory on file <afile>**

(***A <file-név> nevű file kezelésekor elfogyott a memória)

Egy file puffer nem volt lefoglalható. Mivel a tár dinamikusan foglalt, a hibaüzenet megjelenése rendkívüli.

*****Unable to open response file <respfile>**

(***Nem képes megnyitni a válaszfile-t <file-neve>)

A dBCODE feltételezi, hogy a válaszfile tartalmazza saját file specifikációinak legnagyobb részét; abortál, ha nem tudja megnyitni a válaszfile-t.

*****Write error on file <anoutfile> - input line <line#>**

(***Íráshiba a <file-név>-vel megadott file-ban - a <sorszám> input sornál)

A lemez megtelt vagy sérült.

Súlyos hibák

*****<control flow keyword> unexpected in file <afile> on line <line#>**

(***<vezérlésátadási utasítást jelölő kulcsszó>-t nem várt a <file-név> file-ban, a <sorszám> sorban)

Olyan vezérlésátadást talált, mint például az ELSE, ENDIF, LOOP, EXIT, ENDDO, CASE, OTHERWISE, ENDCASE, amelyeknek nincs megfelelő párja - rossz helyen vannak - így nem veszi az utasítást figyelembe. Ezt az üzenetet azonnal követi az elvárható utasítás. Az ENDIF, ENDDO vagy ENDCASE valamelyike fog megjelenni ebben az üzenetben, (amelyikre éppen szükség lenne).

*****<infile> not found!**

(***<input file-név> nem található!)

A dBCODE nem találta a megadott file-t.

*****End of input found prematurely on file <infile> - input line <line#>**

(***Az <input file-név> file idő előtt véget ér - <sorszám> bemenő soron)

A dBCODE egy érvénytelen forrásnyelvi file-t, vagy sérült adathordozót észlelt.

*****Illegal command macro (&) in file <infile> input line <line#>**

(***Illegális parancsmakró az <input file-név> file-ban <sorszám> bejövő soron)

A dBCODE &-MAKRO-val kezdődő parancsot talált.

*****Illegal condition in ON statement, in file <infile> input line <line#>**

(***Illegális feltétel az ON utasításban az <input file-név> file-ban, a <sorszám> számú input sorban)

Hiányzik egy feltétel egy ON utasításból.

*****Illegal DO statement in file <infile> - input line <line#>**

(***Illegális DO utasítás a <file-név> file-ban a <sorszám> számú input sorban)

A dBCODE olyan DO feltétellel találkozott, amelyet nem követ egy CASE, WHILE vagy <file-név>.

*****Illegal RunTime SET option in file <infile> - input line <line#>**

(***Illegális RunTime SET opció a <file-név> file-ban - a <sorszám> számú input sorban)

A dBCODE vagy csak egy SET parancsot talált a sorban, vagy azt egy érvénytelen feltétel követi.

*****Premature end of input on file <infile> - input line <line#>**

(***Idő előtti file vége a <file-név> file-ban - a <sorszám> sorszámú input sorban)

A dBCODE egy érvénytelen forráskódfile-t, vagy sérült adathordozót talált.

*****Read error on file <aninfile> - input line <line#>**

(***Olvasási hiba a <file-név> file-ban - a <sorszám> számú input sorban)

A dBCODE egy érvénytelen forrásfile-t, vagy sérült adathordozót talált.

*****Source - output filename conflict <infile>**

(***Forrás - output file-név konfliktusban vannak <input file-név>-vel)

A forrásfile (.prg) ugyanabban a könyvtárban van, amelyben a kimenő file (.prg). Az output file megnyitása a forráskódot törli. (Ezt a hibát a dBASE, ha a file-ok különböző tartalomjegyzékben vannak, nem mindig veszi észre.)

*****Unrecognized command in file <infile> - input line <line#>**

(***Azonosíthatatlan parancs a <file-név> file-ban - a <sorszám> számú input sorban)

A dBCODE nem tudja azonosítani a parancsot ebben a sorban.

A dBLINKER végzetes és súlyos hibái

Végzetes hibának azt nevezzük, amely a dBLINKER abortálását váltja ki. A súlyos hiba a file aktuális feldolgozásának befejezését okozza. Törli az érvénytelen output file-t, és a következő file-t kezdi feldolgozni. A hibától függően tehát output-unk esetleg nem tartalmazza az összes várt input file-t.

Végzetes hibák

*****Fatal error: can't open file <infile>**

(***Végzetes hiba: nem tudja megnyitni a <file-név> file-t)

Egy input file-t ellenőriztünk, és úgy terveztük, hogy benne lesz az output file struktúrában. Ez a file azonban később eltűnt, így érvénytelen ez az összekapcsolási folyamat.

*****Ran out of memory on file <afile>**

(***A <file-név> file feldolgozásánál kevés lett a memória)

A rendszer nem tudott file puffert lefoglalni. Mivel a helyfoglalás dinamikus, ennek az üzenetnek a megjelenése szokatlan.

*****Read error on file <aninfile>**

(***Olvasási hiba a <file-név> file-on)

A dBLINKER egy érvénytelen file-t, vagy sérült adathordozót észlelt.

*****Unable to open response file <respfile>**

(***Nem képes megnyitni a <file-név> válaszfile-t)

A dBLINKER feltételezi, hogy a válaszfile a saját file specifikációinak nagyobb részét tartalmazza. Abortál, ha nem tudja megnyitni a válaszfile-t.

*****Write error on file <anoutfile> - input line <line#>**

(***Íráshiba a <file-név> file-ban - a <sorszám> számú input sorban)

A lemez megtelt vagy meghibásodott, vagy rosszul adtuk meg a file elérési útját.

Súlyos hibák

*****<infile> is a duplicate!**

(***A <file-név> egy másolat!)

A kiírt file nevének meghatározó része egyezik egy másik input file nevével. Csak az első névdefiníciót fogja a rendszer használni.

*****<infile> is not a dBCODE III file!**

(***A <file-név> nem dBCODE file!)

A dBLINKER egy olyan file-t talált, amelynek formája nem olyan, mint egy dBCODE file-é. Ez a file figyelmen kívül marad.

*****<infile> contains no code!**

(***A <file-név> nem tartalmaz kódot!)

Egy olyan, amúgy érvényes dBCODE input file-t talált, amely nem generál output kódot. Ez a file figyelmen kívül marad.

*****<infile> not found!**

(***<file-név> nem található!)

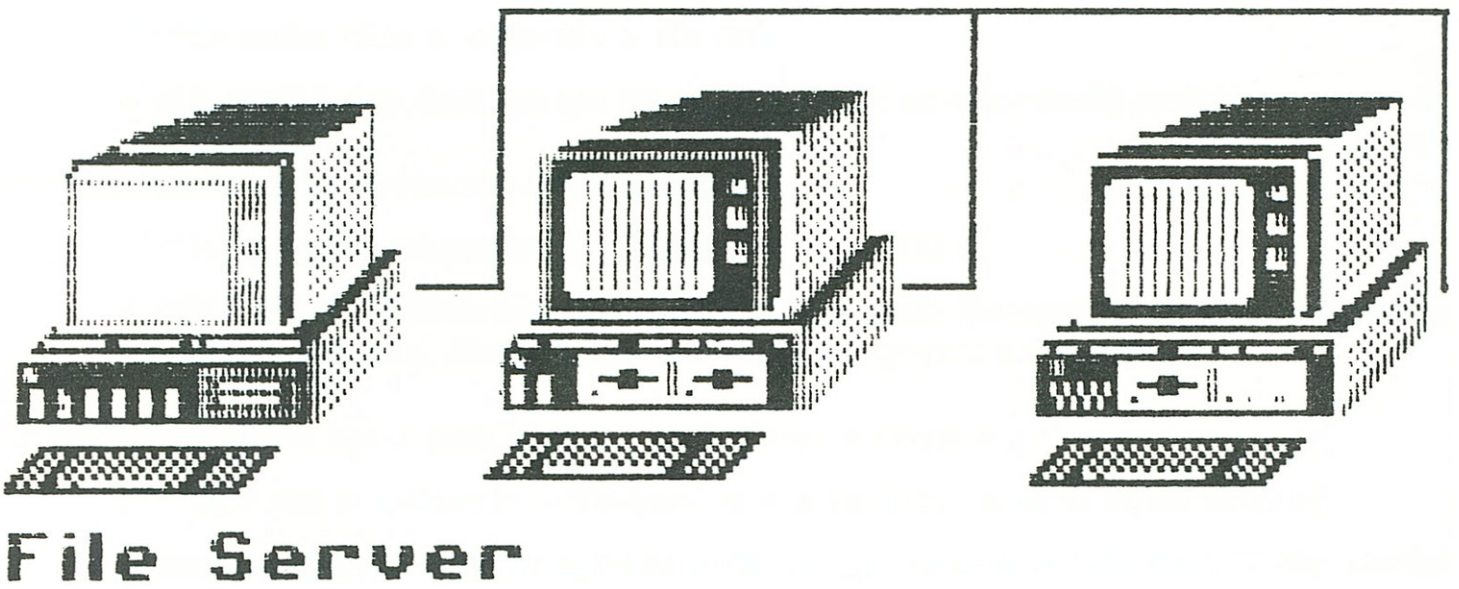
Mivel ez a file nem található, a file-t figyelmen kívül hagyja.

13. A DBASE III PLUS és a lokális hálózatok

13.1. Lokális hálózatok

Az egyfelhasználós (single user) rendszerek megjelenése szorosan kapcsolódik a mikroszámitógépek elterjedéséhez, hiszen a klasszikus számítógépes ügyviteli rendszerek általában valamilyen többfelhasználós (multi-user) rendszerre épültek, mint a mini- vagy a nagyszámítógépek esetén. Mikroszámítógépes környezetben a lokális hálózatok jelentik a tulajdonképpeni továbblépést az egyfelhasználós rendszerektől a komplex többfelhasználós rendszerek felé.

A lokális hálózat több számítógép összekapcsolására épül. A hálózat minden eleme önálló központi egységgel rendelkezik, így az egyfelhasználós rendszerek előnyeit továbbra is élvezhetik a felhasználók. Ugyanakkor a rendszerben lehetőség nyilik a perifériaegységek (lemez,nyomtató stb.) és ami a legfontosab: a programok és az adatok megosztására. Lássunk akkor egy tipikus hálózati rendszert:



13.1.1. ábra. Lokális hálózat

Az állomásokat nagy sebességű adatátviteli vonalak kötik össze. Erre a vonalra csatlakozik minden felhasználó, így ez maga is egy megosztott kommunikációs erőforrás.

A hálózat elemeinek fizikailag a hálózati hardware által megszabott sugarú kör határain belül kell elhelyezkedniük. A hálózati rendszer legalább egyik eleme a közös programok és adatok hozzáférési feltételeit és a hálózati adatkommunikációt vezérli és felügyeli, általában a File Server nevet viseli. Egy hálózati rendszer több File Server-rel is rendelkezhet. A File Serverek vezérelte mikroszámítógépek a munkaállomás (workstation) nevet viselik. A hálózati kommunikáció elsőrendű szempontja az erőforrások, perifériaegységek, adatok és programok megosztása.

Tekintsük át röviden a lehetséges erőforrás megosztási lehetőségeket:

- Software termékek megosztása. Ugyanazt a programfile-t bármelyik hálózati állomás igénybe veheti, így nincs szükség az adott software termék többszörös másolatára.
- Perifériaegységek megosztása. A hálózatban a felhasználók megosztva használhatnak háttértárakat, nyomtatókat stb.
- Adatfile megosztás. Több felhasználó hozzáférhet ugyanahhoz az adatfile-hoz. Az adatredundancia megszüntethető.

13.1.1. A lokális hálózatok és a dBASE III Plus

A dBASE III PLUS adatbáziskezelő rendszer a lokális hálózatok két típusán működtethető. Ezek az IBM PC Network és a Novell Netware lokális hálózat. Mindkét hálózat DOS 3.1. vagy ennél magasabb verziószámú operációs rendszerben működik.

Az IBM PC Network hálózat esetén a hálózati software: IBM PC NETWORK PROGRAM szorosan kapcsolódik az IBM PC hálózati hardware-hez és csakis ezen működik. Egyszóval az IBM PC hálózat hardware specifikus rendszer. A hálózati programcsomag elemei:

- a munkaállomásokon a DOS 3.1 operációs rendszer
- a file server program
- a redirektor program
- server segédprogramok

A file server program tulajdonképpen egy DOS 3.1 rendszerben futtatható alkalmazás.

A redirektor a hálózati parancsok operációs rendszer szintről történő kiadását teszi lehetővé.

A Novell hálózati programcsomag úgyszólván hardware független. Jelenleg 21 különböző hálózati hardware-on működik. A hálózati software használata és működése az összes hálózati hardware-on azonos.

A Novell három féle hálózati programcsomagot hozott forgalomba a file serverként működő számítógép különböző típusainak megfelelően:

- Novell Advanced Netware/86 IBM PC/XT és IBM PC/AT file server számára
- Novell Advanced Netware/286 IBM PC/AT file server számára
- Novell Advanced Netware/68, a 68000-as mikroprocesszorra épülő file server-ek számára

A Novell Advanced Netware hálózati programcsomag elemei a következők:

- DOS 3.1 vagy ennél magasabb verziószámú operációs rendszer
- redirektor program
- file server software
- hálózati segédprogramcsomag

Az IBM PC Network rendszertől eltérően, a file server software nem DOS 3.1 alkalmazás. A software a file server erőforrásait közvetlenül vezérli és ellenőrzi. A Novell Netware tehát hálózati operációs rendszer. A Netware vezérelte lemezeket, a Netware hálózati programcsomag segítségével kell formáznunk. A Netware formázta lemezek nem kezelhetők DOS lemezekként.

A Netware file server önálló munkaállomásként is működhet miközben a file server feladatait is ellátja. Az ilyen, úgynevezett nem-dedikált módban működő file server használata nem a legszerencsésebb, sőt kifejezetten ellenjavallt. Több felhasználó hozzáférési igényeinek és munkaállomásként egy felhasználó számításainak egyidőben megfelelni, ez csakis a hálózati teljesítmény, így pedig minden felhasználó rovására történhet.

Mig az IBM PC Network működése a DOS funkciók használatára korlátozódik, addig Novell Netware a DOS funkciók mellett, a hálózati funkciók gazdag választékát szolgáltatja.

Összevetve az előbbiekben röviden vázolt két hálózati rendszert, az előnyöket mérlegelve a Novell Advanced Netware hálózati rendszer széleskörű elterjedésének egyszerű a magyarázata. A továbbiakban a dBASE III PLUS működését elsősorban Novell hálózati rendszerben tárgyaljuk, mivel ma Magyarországon is ez a legelterjedtebb lokális hálózati rendszer.

A dBASE III PLUS hálózati működésének feltételei:

- DOS 3.1 vagy ennél magasabb verziószámú operációs rendszer
- merevlemezes háttértárral és legalább 640 KB központi memóriával rendelkező file server
- legalább egy hajlékonylemezes háttértárral és 384 KB központi memóriával rendelkező munkaállomások

13.2. A dBASE III PLUS hálózati programcsomag

13.2.1. A dBASE III PLUS programcsomag elemei

- dBASE ADMINISTRATOR, az alap dBASE III PLUS hálózati funkciókkal bővített változata
- dBASE ACCESS, a munkaállomásokról a file serveren installált dBASE ADMINISTRATOR elérését szolgáló program.
- PROTECT, a dBASE adatvédelem felépítését szolgáló program

13.2.2 Telepítési követelmények

A dBASE ADMINISTRATOR-t a file server-re kell telepíteni. Összetevői az alap dBASE III PLUS és a dBASE hálózati adminisztrációs rendszer. A dBASE ADMINISTRATOR-t nyilvánvalóan egy megosztható minden felhasználó által hozzáférhető könyvtárba kell bemásolni.

A dBASE ACCESS teszi lehetővé az állomások számára a file server-en installált dBASE ADMINISTRATOR elérését. A dBASE ACCESS program, annyi különböző kópiája szükséges, ahány munkaállomáson használni kívánják az adatbáziskezelő rendszert. Ugyanarról a dBASE ACCESS lemezről több munkaállomás dBASE ADMINISTRATOR-hoz kapcsolása nem lehetséges. Az hálózatba kapcsolódás folyamata a dBASE ADMINISTRATOR és a dBASE ACCESS kölcsönhatására épül. Ha a file server maga is munkaállomásként működik, a file server felhasználójának is rendelkeznie kell a dBASE ACCESS program külön kópiájával.

A PROTECT segédprogram használata opcionális. Az adatvédelmi rendszer létrehozásának és folyamatos karbantartásának eszköze. Elejét veszi az adatbázis file-ok jogosulatlan hozzáférésének.

13.2.3. A dBASE ADMINISTRATOR telepítése

A telepítéshez szükséges lemezek:

- a dBASE ADMINISTRATOR #1
- a dBASE ADMINISTRATOR #2

Az adminisztrátor telepítése a file server egy könyvtárába történik, amely minden dBASE felhasználó számára hozzáférhető. A dBASE ADMINISTRATOR általában másolási védelem nélküli lemezeken kerül forgalomba, telepítése az INSTALL.BAT indításával történik. A dBASE ADMINISTRATOR lemezterület igénye minimálisan 2.5 Mbyte.

IBM PC Network lokális hálózat esetén a file server-en és a munkaállomásokon a következő műveleteket kell még végrehajtani:

- a file serveren, amelyre a dBASE ADMINISTRATOR-t telepítettük:

  C > NET SHARE DBAD = C:\DBASEP /RWC

  DBAD a dBASE ADMINISTRATOR keresését szolgáló rövid név.
  DBASEP a dBASE ADMINISTRATOR-t tartalmazó alkönyvtár neve.

  Az előbbi parancs nyomán a DBASEP könyvtár a munkaállomások számára hozzáférhetővé válik.

- a munkaállomásokon:

  A DBASEP alkonyvtár itt mint külön meghajtó jelentkezik (például "D").

  A > NET USE D:\\FSNEV\DBAD

  FSNEV a file server nevét jelöli.

13.2.4. A dBASE ACCESS telepítése

Bár a forgalmazó minden bizonnyal mellékel a dBASE ACCESS lemezekhez kezelési útmutatót, hasznos lehet az installálás feltételeinek és folyamatának rövid összefoglalása.

A dBASE ACCESS program indítása, vagyis a file server-re telepített dBASE ADMINISTRATOR elérése a munkaállomásokról, állomásonként legalább 384K központi memóriát, DOS 3.1 vagy magasabb verziószámú operációs rendszert és megfelelő rendszerkonfigurációs file-t (config.sys) követel.

A rendszerkonfigurációs file-nak (config.sys) a következő sorokat kell tartalmaznia:

```
files = 24
buffers = 20
lastdrive = x (x A-N közötti betű)
```

Ha a CONFIG.SYS file-t változtattuk, az operációs rendszert újra kell indítanunk a megfelelően kialakított rendszerkonfigurációs file-val.

A dBASE ACCESS program általában másolásvédetten kerül forgalomba. Indítása a munkaállomásokon általában a dBASE ACCESS lemezről történik, ugyanakkor a dBASE ACCESS a munkaállomás merevlemezes tárolójára is telepíthető. A telepítés előfeltétele legalább 2.5 Mbyte szabad hely a merevlemezes tárolón. A telepítést követően a dBASE ACCESS a munkaállomás merevlemezes meghajtójáról indítható.

A dBASE ACCESS telepítése a munkaállomás merevlemezes háttértárára a következőképpen történik:

- "A" jelöli a meghajtót, amelyben a dBASE ACCESS lemez található
- "C" a munkaállomás merevlemezes meghajtója
- A > INSTALL C:ACCESS program futtatása és az intallációs képernyő utasításainak pontos betartása után a program sikeres telepítést jelző üzenettel tér vissza.
- A dBASE ACCESS lemez a továbbiakban, csak a merevlemezes tároláról történő visszatelepítés után használható. A visszatelepítés a következőképpen történik:

  A > UNISTAL C:ACCESS

  A visszatelepítési képernyő utasításainak betartása után, ha ugyanazt a lemezt helyeztük az "A" meghajtóba, mint telepítéskor a visszatelepítés sikeresen megtörténik, és a program ennek megfelelő üzenettel tér vissza. Ha nem a megfelelő dBASE ACCESS lemezt használjuk, a program hibaüzenet megjelenítése után befejeződik.

A dBASE ACCESS használata

Az ACCESS indítása a következőképpen történik:

A > PATH = D:\DBASEP	- a dBASE ADMINISTRATOR könyvtár elérhetővé a PATH DOS parancs segítségével
A > ACCESS M = D:\DBASEP	- dBASE ACCESS indítása

"A" a munkaállomás hajlékonylemeze meghajtóját jelöli, amelybe a dBASE ACCESS lemezt helyeztük. DBASEP a dBASE ADMINISTRATOR-t tartalmazó alkönyvtár neve. A "C" merevlemezes meghajtóról az előbbiekben vázolt módon installált dBASE ACCESS hasonlóképpen indítható. A pont prompt vagy PROTECT-tel védett rendszer esetén csoportnév, felhasználónév és jelszó bevitelét váró képernyő megjelenése a dBASE ADMINISTRATOR sikeres indítását jelzi.

A dBASE ADMINISTRATOR elérésével egyidejűleg dBASE alkalmazás a következőképpen indítható:

A>ACCESS:alknev M=D:\DBASEP

Az "alknev" alkalmazásnak az dBASE ADMINISTRATOR-ral azonos könyvtárban kell lennie.

IBM PC Network lokális hálózat esetén, ha a NET USE parancs tartalmazza a dBASE ADMINISTRATOR keresését szolgáló rövid nevet, minden további dBASE ADMINISTRATOR keresési út megjelölés fölöslegessé válik.

13.3. Adatvédelem, adatvédelmi szintek, dBASE adatvédelem

A hálózati adatbáziskezelő rendszer elsőrendű feladatai közé tartozik az adatvédelem és biztonság. Az adatbázis a rendszer megosztva használt eleme, s mint ilyen sebezhető. A rendszer adatvédelmi funkciói hivatottak védeni az adatbázis file-jait az illetéktelen hozzáféréstől, olvasástól, változtatástól.

13.3.1. Novell/Netware LAN hálózati adatvédelem

A védelmi rendszer első (külső) gyűrűjét maga az operációs rendszer képezi.

13.3.1.1. A hálózatba kapcsolódás

Novell/Netware esetében minden egyes állomásnak (user) egy file-serverhez kell kapcsolódnia. A bejelentkezés amellyel itt találkozunk már a többfelhasználós rendszerekkel mutat hasonlóságot: minden egyes állomás bejelentkezési névvel és egy ehhez opcionálisan rendelt jelszóval rendelkezik. A jelszót maga a felhasználó is változtathatja.

A hálózatszervezés a SUPERVISOR nevet viselő különleges jogokkal felruházott Novell/Netware által minden esetben felismert és hálózatba kapcsolt felhasználó segítségével történik. Ez a felhasználó a hálózati adatadminisztrátor szerepét tölti be.

13.3.1.2. Könyvtárhozzáférés

A hálózat felépítése során a felhasználóknak egyes könyvtárakban adott hozzáférési jogokat kell biztosítanunk. Az adatadminisztrátor (SUPERVISOR) vagy adminisztrá-

tori jogokkal felruházott felhasználó könyvtár- és file-hozzáférési jogokat egyaránt korlátozhat.

A Novell/NetWare hozzáférési jogok nyolc típusát ismeri

- Olvasási jog: az adott könyvtárban olvasási joggal rendelkező felhasználó olvashatja bármely nyitott file adtait. Jele: R.
- Írási jog: a felhasználó adatokat írhat bármely nyitott file-ba abban a könyvtárban, ahol ezzel a joggal rendelkezik. Jele: W.
- Megnyitási jog: ezzel a joggal felruházott felhasználó megnyithat minden file-t, amely az adott könyvtárhoz tartozik. Ez utóbbi jog nélkül a felhasználónak sem olvasási sem írási lehetősége nincs az adott könyvtárban létező file-ok esetében. Jele: O.
- Létrehozási jog: ezzel a joggal felruházott felhasználó file-okat hozhat létre az adott könyvtárban. Alkönyvtár létrehozásához, a "parentális" jog mellett erre a jogra is szükség van. Jele: C.
- Törlési jog: ezzel a joggal rendelkező felhasználó az adott könyvtárban file-okat törölhet. Alkönyvtár törléséhez a parentális jog mellett erre a jogra is szükség van. Jele: D.
- "Parentális" jog: ez a jog alkönyvtárak létrehozását, törlését és átnevezését teszi lehetővé. Jele: P.
- Keresési jog: ez a jog az adott könyvtárban keresést biztosít, a felhasználónak lehetősége van a könyvtárfile-ok megjelenítésére, azaz pl. a DIR parancs kiadására. Jele: S.
- File attributumok módosítási joga (jele A): ez a jog egy adott könyvtárban a file-attributumok és hálózatra jellemző kiterjesztett file információk megváltoztatásának lehetőségét biztosítja. Öt file attributum állítható be a FLAG Netware paranccsal. Ezek következők:

- csak olvasható (Read Only) jele: R/O
- írható/olvasható (Read/Write) jele: R/W
- Megosztható (Shareable) jele: S
- Nem megosztható (Non Shareable) jele: NS
- Normál N jele: N

Alapállapotban egy file a hálózatban normál attributummal rendelkezik. Ez írható/olvasható és nem megosztható tulajdonságot takar.
A módosítási jog ugyanakkor egy file átnevezési lehetőségét és a Parentális joggal együtt könyvtár átnevezési lehetőségét is biztosít.

13.3.2. dBASE III Plus adatvédelem — Alapfogalmak

A védelmi rendszer második gyűrűjét a DBASE III PLUS védelmi rendszere képezi. A jelszóvédett PROTECT.EXE program a file-ok titkos, kódolt változatának előállításával, és az úgynevezett felhasználói profilok kialakításával látja el adatvédelmi feladatát. A felhasználói profil egy adott felhasználó hálózati jellemzőinek gyűjteménye, amely a felhasználó adathozzáférési lehetőségeit szabályozza.

A PROTECT használata opcionális, installálása nem kötelező, de egyszeri használata után az adatbázis file-ok hozzáférése a védelmi rendszer ellenőrzése alá kerül.

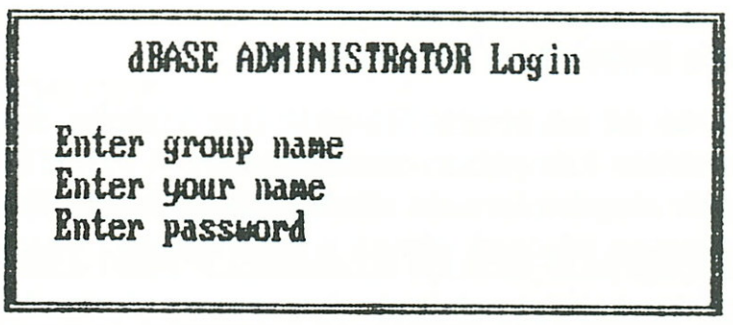
A PROTECT az adatvédelem három tipusát ismeri, ezek a következők:

- A bejelentkezés védelme. A PROTECT adatvédelmi rendszer installálása után minden programindítást (Runtime+), illetve a pont jelzés (prompt) megjelenését egy bejelentkezési eljárás előzi meg.
- Az adattitkosítás, kódolás. A titkosítás adott algoritmus szerint, kulccsal történik. Minden egyes adatbázis file-t a PROTECT egy és csakis egy felhasználói csoporthoz (group) rendel. Az adatkódolás kulcsa szorosan kapcsolódik a felhasználói csoport nevéhez és csoportonként különböző. A csoport neve alapján történik a titkosított file dekódolása (az adott csoporthoz tartozó felhasználói állomásokon).
- A file és mező hozzáférések ellenőrzése. A PROTECT minden felhasználóhoz egy hozzáférési szintet, és minden file-hoz és mezőhöz, úgynevezett "privilégiumokat", előjogszinteket rendel. Ezen értékek sajátos viszonya szabályozza a file és mező hozzáféréseket.

13.3.2.1. A bejelentkezési védelem és a DBSYSTEM.DB file

13.3.2.1.1. A bejelentkezési védelem

A PROTECT védelem első szintje a bejelentkezési védelem. A Runtime+ programindítást és a dBASE pont prompt megjelenését, a védelmi rendszer installálása esetén egy bejelentkezési képernyő előzi meg. A képernyő három adat begépelését várja. Ezek a felhasználói csoport, a felhasználó neve, valamint a jelszó, amint ez az alábbi ábrán látható.



13.3.1. ábra: Bejelentkezési képernyő

Mindhárom már előbb említett adat meghatározására a PROTECT installálása közben kerül sor, mint azt a későbbiekben látni fogjuk. A három bejelentkezési adat megadá-

sában a kis és nagy betűk egyenértékűek. A jelszó begépelés közben természetesen nem jelenik meg a képernyőn.

A Runtime+ programindítás vagy az dBASE ADMINISTRATOR (DBA.EXE) betöltése a PROTECT által létrehozott file keresésével indul. Ha ez a file létezik, akkor kerül sor a bejelentkezési képernyő megjelenítésére. Ha a bejelentkezési adatok megegyeznek a PROTECT által a file-ba felvett valamelyik felhasználó azonos adataival, akkor a rendszer elfogadja a bejelentkezést, és a csoporthoz rendelt titkosítási kulcs (system public key), valamint a felhasználóhoz rendelt hozzáférési szint a továbbiakban a rendszer rendelkezésére áll.

Helytelen bejelentkezés esetén, a rendszer még kétszer felkínálja a javítás lehetőségét, mielőtt hibajelzéssel visszatérne az operációs rendszerhez. A bejelentkezési folyamat az "Esc" billentyű leütésével bármikor megszakítható.

13.3.2.1.2. A DBSYSTEM.DB file

A PROTECT által létrehozott és karbantartott DBSYSTEM.DB file a felhasználók bejelentkezési és hozzáférési file-ja. Ez a file kódolt, tartalma nem olvasható, és nem is dekódolható. A DBSYSTEM.DB file alapvető jellemvonásait az alábbiakban összegezzük:

- a DBSYSTEM.DB rekordjai az egyes felhasználók hálózati jellemzőit tömörítő ún. felhasználói profilok. Egy rekord a felhasználó bejelentkezési nevét, a jelszavát, a felhasználói csoport nevét és a felhasználóhoz rendelt hozzáférési szintet tartalmazza.
- a DBSYSTEM.DB változtatása és az új változat elmentése a file változtatás előtti állapotának elmentését is eredményezi. Az elmentett változat neve DBSYSTEM.BAK.
- a DBSYSTEM.DB törölhető és felülírható. A PROTECT-tel több DBSYSTEM.DB file, tehát több védelmi rendszer is létrehozható. Az aktuális DBSYSTEM.DB-nek a DBA (dBASE ADMINISTRATOR)-ral, vagy RUNTIME+ programmal kell közös könyvtárban szerepelnie.

Megjegyzés: A DBSYSTEM.DB file felülírása nem érinti a file hozzáférési rendszert és az adatkódolás/dekódolás rendszerét sem.

13.3.2.2. Adatkódolás (titkosítás) és dekódolás

Az adattitkosítási eljárás az adatbázis file-okat úgy kódolja, hogy azok tartalma a titkosítás kulcsának ismerete hiányában olvashatatlan. A PROTECT-tel védett adatbázis file-ok, és index file-jaik alapértelemzés szerint titkosítottak. A titkosított adatbázis file index file-jai automatikusan titkossá válnak a legközelebbi használatkor.

13.3.2.2.1. A kódolás (titkosítás) kulcsa és a csoport (group) fogalma.

A titkosított file a titkosítás kulcsának felhasználásával válik olvashatóvá. A helyes titkosítási kulcs, bármelyik, a felhasználóval megegyező csoporthoz tartozó file titkosságának feloldására képes. Ezt a kulcsot, mint azt már a bejelentkezési védelemnél em-

lítettük, bejelentkezéskor állítja elő a rendszer. Ugyanott találkoztunk a felhasználói csoport fogalmával, amely szorosan összefonódik a titkosítási eljárással. Mint a későbbiekben azt látni fogjuk, a file előjog rendszer felépítésének része a csoport nevének megjelölése, amelyhez a szóbanforgó file tartozik. A védett file hozzáférése csak a felhasználói és file csoport azonossága esetén válik lehetővé.

A védelmi rendszer egyszerűsítése és áttekinthetősége érdekében javasolt az összes felhasználó és file-ok egy és ugyanazon csoporthoz rendelése. A mező hozzáférési szintek és felhasználói hozzáférési szintek segítségével majd úgyis további korlátokat emelhetünk az illetéktelen hozzáférés útjába.

Az adatbázis file dekódolása a felhasználó állomásoknál sikeres bejelentkezés után automatikusan megtörténik a felhasználó és file csoportazonosság alapján.

Az előbbiek ismeretében tehát:

- ahhoz, hogy egy felhasználó egy adott csoporthoz tartozó file-hoz hozzáférhessen az adott csoport tagjaként kell bejelentkeznie;
- a csoport nevének megváltoztatása esetén a változtatás előtti csoporthoz rendelt file-ok hozzáférhetetlenné válnak, éppen ezért a csoportnév változtatása nem ajánlott;
- egy adott csoport törlése, a csoporthoz rendelt file-ok titkosítatlan formában történő kimentése előtt, az adott file-ok hasznavehetetlenségéhez vezet. Hozzáférésük lehetetlenné válik. A titkosítatlan formában történő kimentés a SET ENCRYPTION utasítás segítségével történik;

13.3.2.2.2. A SET ENCRYPTION utasítás (titkosítás be/ki kapcsolása)

A felhasználó, a számára hozzáférhető titkosított file-ok másolatainak titkossága fölött szabadon rendelkezhet. Ha az adatbázis rendszer a PROTECT védelme alatt áll az egyben azt is jelenti, hogy a titkosítás bekapcsolt állapotban van. Az ilyen körülmények között létrehozott file-ok mindig titkosak. Az adattitkosítás azonban kikapcsolható. A titkosítás kikapcsolása után létrehozott másolat file-ok már nem lesznek titkosak.

A titkosítás ki- és bekapcsolásának eszköze a SET ENCRYPTION utasítás. A SET ENCRYPTION ON parancs a titkosítást be-, a SET ENCRYPTION OFF utasítás a titkosítást kikapcsolja.

Példa:
```
USE FILE1
SET ENCRYPTION OFF
COPY TO FILE2
```

A FILE2 másolati file már nem lesz titkosított.

A SET ENCRYPTION utasítással kapcsolatos további információk hálózati környezetben használható utasítások és parancsok részletes leirásának szentelt fejezetben állnak rendelkezésére.

13.3.2.3. File és mezőhozzáférés

13.3.2.3.1. File előjogok (File privileges)

Miután az előbbiek ismeretében a titkosított file megnyitása megtörtént, a file-lal kapcsolatos minden további tevékenység szabályozását a felhasználóhoz rendelt hozzáférési szint és a filehoz rendelt hozzáférési előjogok irányítják.

A felhasználói hozzáférési szint tulajdonképpen kategóriákba sorolja a felhasználókat, hiszen azonos hozzáférési szinttel rendelkező felhasználók azonos előjogkörrel rendelkeznek. A továbbiakban a felhasználói hozzáférési szint és a felhasználói kategória mint egyenértékű fogalmak szerepelnek.

A hozzáférési szintek fokozatait 1 és 8 közötti természetes szám jelöli. Így nyolc különböző felhasználói kategória kijelölésére nyilik mód.

A felhasználói kategóriák és file előjogok viszonya engedélyezi vagy tiltja a file hozzáféréseket. Az adatbázis file előjogainak tábláját az index file-ok is megkapják első megnyitásuk alkalmával.

A file előjogok a következők:

- Új rekordok filehoz fűzése (APPEND) – bővítési előjog
- Rekord törlés (DELETE) – törlési előjog
- Rekord olvasás – olvasási előjog
- Rekord változtatás (REPLACE, EDIT, INSERT) – módosítási előjog

A file előjogokhoz 1 és 8 közötti természetes szám rendelhető. Minden felhasználó, amelynek hozzáférési szintje (kategóriaszáma) kisebb vagy egyenlő a file előjoghoz rendelt számnál, rendelkezik az adott a előjoggal.

Ebből rögtön következik hogy az 1.kategóriába sorolt felhasználó a legtöbb (minden) előjog birtokosa, mig a 8. kategoria felhasználója rendelkezik a legkevesebb előjoggal. Ugyanakkor a file előjogok 1-es fokozata a legkorlátozóbb, a 8-as fokozat pedig a legkevésbé korlátozó hatású.

13.3.2.3.2. Mező előjogok (Field privileges)

Működési elvük alapvetően különbözik a file előjogokétól. Minden felhasználói szinthez külön-külön, a mező előjogok alábbi három szintjét rendelhetjük:

- TELJES körű (FULL) – irás/olvasás; jele: F
- CSAK OLVASÁS (R/O) – olvasás; jele: R/O
- NINCS (NONE) – sem olvasás, sem írás; jele: N

Alapértelmezés szerint a mezők írhatók/olvashatók, tehát a mező előjogok **teljes** körűek (FULL tipus).

Lássuk, ezek után hogyan is működnek ezek a előjogok.

A szélesebb körű (file) előjogok mindegyikéhez nyolc különböző szintet rendelhetünk egy 1 - 8 közötti számmal, mint azt már az előbbiekben említettük. Például, ha egy

file-hoz 3 -as olvasási szintet rendelünk, a 3 és ennél kisebb hozzáférési szinttel rendelkező felhasználók olvashatják az adott file-t. A 4 és annál nagyobb kategoriájú felhasználók számára a file olvashatatlan.

A mező előjogokkal tovább korlátozhatjuk a hozzáférések lehetőségét. Például ha egy adott mező olvasását csak a 3-as és 6-os kategóriába sorolt felhasználók számára akarjuk engedélyezni, a következőképpen kell eljárnunk:

1. A file-hoz 6-os olvasási előjogszintet rendelünk.
2. Az 1,2,4,5 felhasználói kategoriákhoz NINCS(NONE) mezőprivilégiumot társítunk.
3. A 3-as és 6-os kategoriák esetében a mező vagy TELJES (FULL) vagy CSAK OLVASHATÓ (R/O) privilégiumot kap.

A file előjogok hatásköre a mező előjogok hatáskörénél szélesebb. Igy az előbbi példánál, a 7-es és 8-as felhasználói kategóriák számára nincs szükség mezőhozzáférés beállításra. Ha file olvasási előjog 6-os fokozatú, akkor a 6-nál nagyobb számmal jelölt hozzáférési szinttel rendelkező felhasználókat ez megfosztja a file olvasás, így minden egyes mező olvasásának jogától is.

File/mező előjogok és felhasználói szintek kapcsolatát táblázatokkal tehetjük szemléletesebbé. Az alább vázolt táblázatok kitöltése a hálózati rendszerszervezés szerves része.

File előjogok

Előjogszint	1	2	3	4	5	6	7	8
Bővítés	*							
Törlés				*				
Olvasás								*
Változtatás	*							

A "*" az előjogszint és a file előjog metszéspontjába kerül. Így fentebb, az adott file minden felhasználó számára olvasható, hiszen a legkevésbé korlátozó (8-as) előjogszinttel rendelkezik. A bővítés az 1., a törlés az 1.,2.,3.,4. a változtatás az 1. hozzáférési szinttel felruházott felhasználók számára engedélyezett.

Mező előjogok

	Felhasználói hozzáférési szint							
Mező neve	1	2	3	4	5	6	7	8
Mező_1	.	.	.	T	.	.	.	.
Mező_2	.	.	.	O	.	.	.	.
Mező_3	.	.	.	N	.	.	.	.
.								
.								
Mező_n	.	.	.	T	.	.	.	.

A mezőnév és a felhasználói hozzáférési szint (kategória) metszéspontjába a mező-előjogszintet meghatározó kulcsszó kezdőbetűje kerül. Fentebb a 4-es felhasználói

szinthez rendeltünk mezőelőjogszinteket. "T" teljes előjogkört, "O" olvasási előjogkört, míg "N" a mezőelőjogok hiányát jelöli.

13.3.2.3.3. A titkosítás és az előjogok

A file előjog - felhasználói kategória tábla felépítése minden egyes adatbázis file számára a PROTECT feladata. Ugyanaz a PROTECT menü funkció felépíti a file előjog táblát és titkosítja az adatbázis file-t. Az adat titkosítás automatikus, a file előjog tábla elmentésével egyidőben történik. A PROTECT ekkor létrehozza az adatbázis file titkosított változatát .CRP kiterjesztéssel, ugyanakkor a file titkosítatlan változata is megmarad .DBF kiterjesztéssel. Amit még tenni kell a védelem működtetésének érdekében a következő pontokban foglalható össze:

- töröljük az adott könyvtárból a .DBF kiterjesztésű titkosítatlan változatot; a .DBF file kimentése javasolt, hiszen esetleg szüksége lehet az adatfile titkosítatlan változatára;
- nevezzük át a .CRP kiterjesztésű titkosított változatot .DBF kiterjesztésűvé;

13.3.3. A védelmi rendszer létrehozása. A PROTECT használata.

Az előbbiek ismeretében, a PROTECT segítségével nekiláthatunk, a DBASE III PLUS rendszerünk védelmének felépítéséhez.

13.3.3.1. PROTECT használati alapismeretek

13.3.3.1.1. A PROTECT indítása, adminisztrátor jelszó meghatározása

A PROTECT az alábbi bejelentkezési képernyővel indul:

```
dBASE ADMINISTRATOR Password Security System

Enter administrator password
```

13.3.2. ábra. PROTECT bejelentkezési képernyő

Ez a PROTECT bejelentkező képernyője. A hálózatkezelőt (az adminisztrátort) a jelszó megadására szólítja föl.

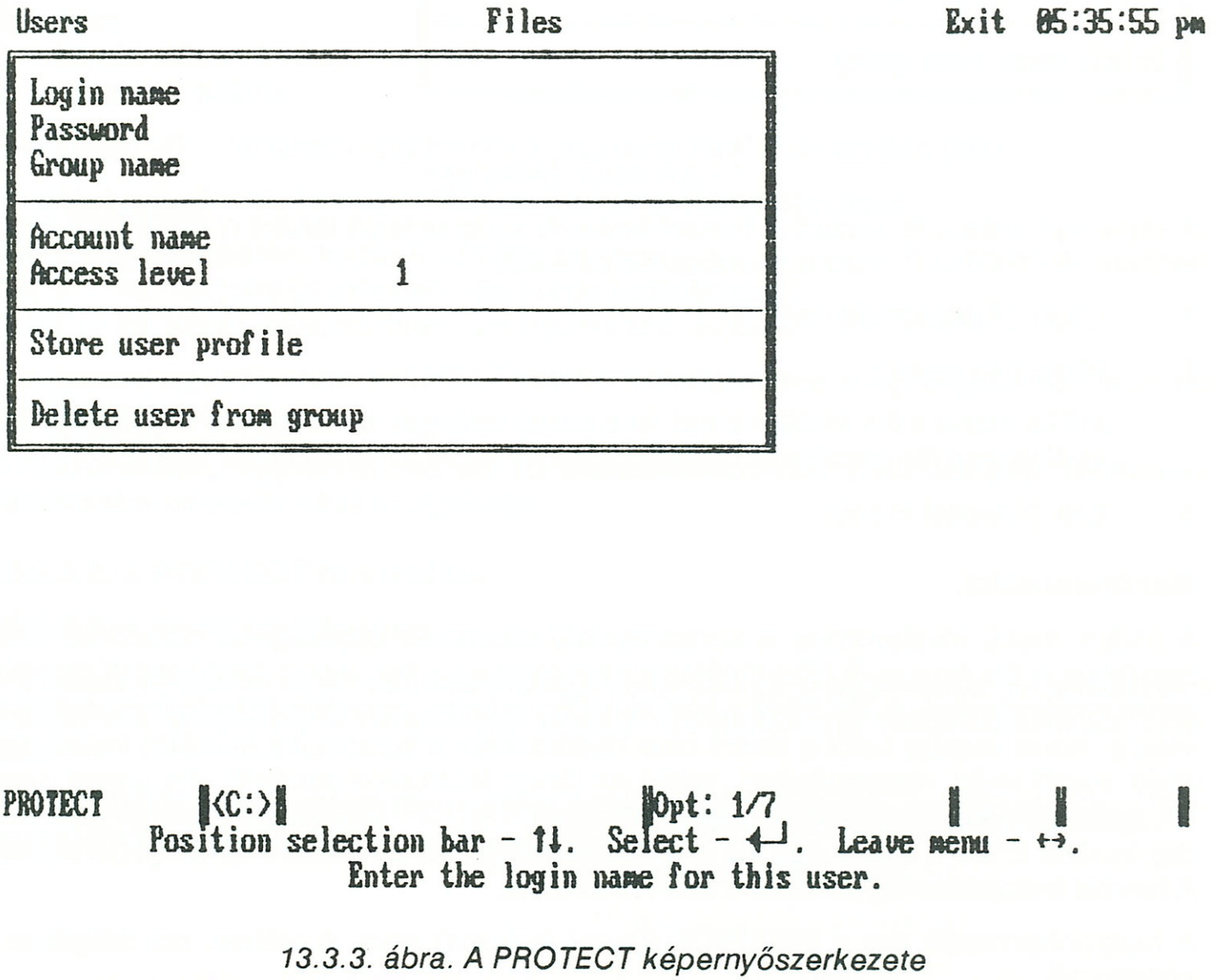
Első indításkor, pontosabban ha a PROTECT nem találja a DBSYSTEM.DB file-t a jelszó kétszeri megadására van szükség, vagyis az először megadott jelszó érvényességét meg kell erősíteni. Minden későbbi indítás alkalmával a jelszót csak egyszer kell megadni. Érvenytelen jelszó esetén a PROTECT "Unauthorized login" hibajelzéssel

befejeződik. Igy van ez első indításkor is, ha az először megadott jelszó nem egyezik a másodszor megadott jelszóval.

A jelszó nem jelenik meg a képernyőn, a kis és nagy betűk egyenértékűek. A fokozott biztonság érdekében a maximálisan megengedett 16 karakter hosszú jelszó használata javasolt. Hangsúlyozzuk a jelszó megőrzésének a fontosságát. A védelmi rendszer változtatása csak az adminisztrátor jelszó ismeretében lehetséges.

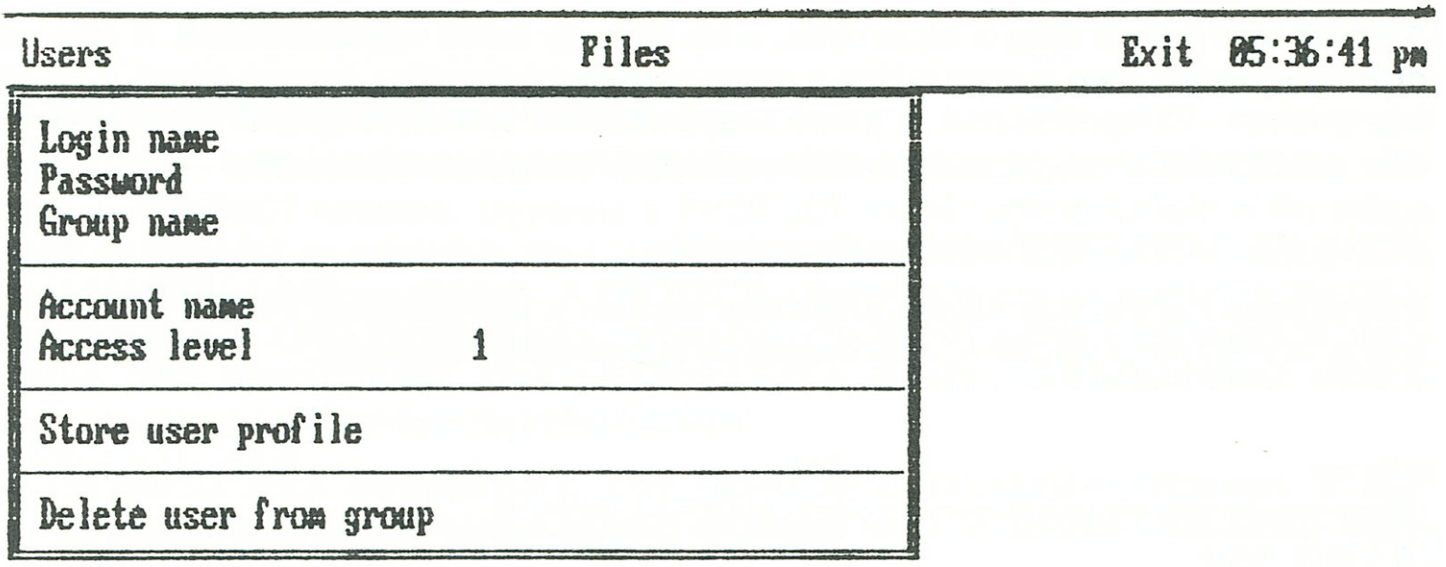
13.3.3.1.2. A PROTECT képernyőszerkezete

Helyes jelszó megadása esetén engedélyt kapunk a menüvezérelt PROTECT használatára. A PROTECT képernyőszerkezetét az alábbi ábra illusztrálja.



13.3.3. ábra. A PROTECT képernyőszerkezete

A képernyő első sora az úgynevezett menü sor. Adott menü kiválasztása az inverz kijelzésű mező megfelelő tételre mozgatásávak, majd az ENTER leütésével érhető el.



13.3.4. ábra. Menü sor

A képernyő második sorától a huszonkettedik sorig terjedő terület a PROTECT menü területe. A PROTECT három menüvel rendelkezik:

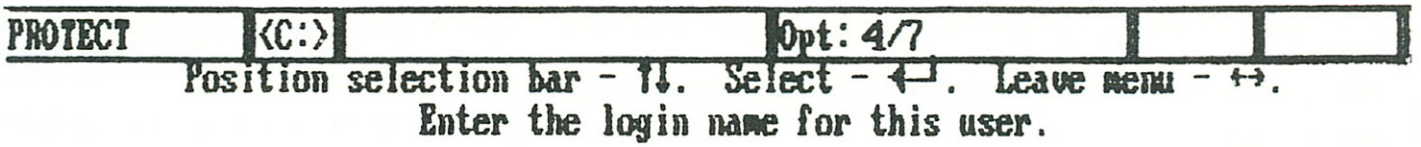
1. Users (Felhasználó menü)
2. Files (File menü)

   A File menü a file kiválasztását és a mező előjogok beállítását a DBF file-ok és a kiválasztott file mezőinek megjelenítésével segíti. (File és mező listák)
3. Exit (Kilépési menü)

Menühasználat:

A kívánt menü kiválasztása a kurzorvezérlő billentyűk segítségével érhető el. A file menüben a file és mező (Field) tételek kiválasztása a file, illetve mező listák megjelenését eredményezi. A listákból a File menübe a balra mozgató billentyűvel térhetünk vissza. Adott menün belül a kívánt tétel kiválasztása a kurzorként működő inverz kijelzésű mező le-fel mozgatásával, majd az Enter leütésével történik. Az inverz mező mozgatását a kurzort le- fel mozgató billentyűk végzik . Információ bevitele előtt mindig üssünk Enter-t, ekkor egy kis nyil megjelenése jelzi az adatbevitel engedélyezését. A bevitel befejezése ugyancsak Enter-rel történik.

A huszonharmadik sor a PROTECT állapot (státusz) sora. A státusz sor felépítése a következő:



13.3.5. ábra. Státusz sor

Az első,a műveleti szakasz, amely mindig a PROTECT szöveget tartalmazza. A második szakasz az aktuális meghajtó kijelzését szolgálja. A harmadik szakaszt a PROTECT nem használja. A negyedik szakasz a kiválasztott menün, vagy listán belül kiválasztott tétel sorszámát és a maximálisan kiválasztható tételek számát jelzi a következő formában:

Opt: n/m
n – az aktuálisan kiválasztott tétel
m – a tételek száma

Tehát Opt: 4/7 – a lehetséges hétből a negyedik tétel kiválasztását jelzi.

Az utolsó két státusz szakasz a billentyűzet állapotáról tájékoztat:

INS – ha Beszúrás módban van (Ins leütése)
CAPS – ha Nagybetűs módban van (Caps Lock leütése)
NUM – ha Numerikus módban van (Num Lock leütése)

A huszonnnegyedik sor a PROTECT menük használatához nyújt segítséget. Tartalma mindig ugyanaz.

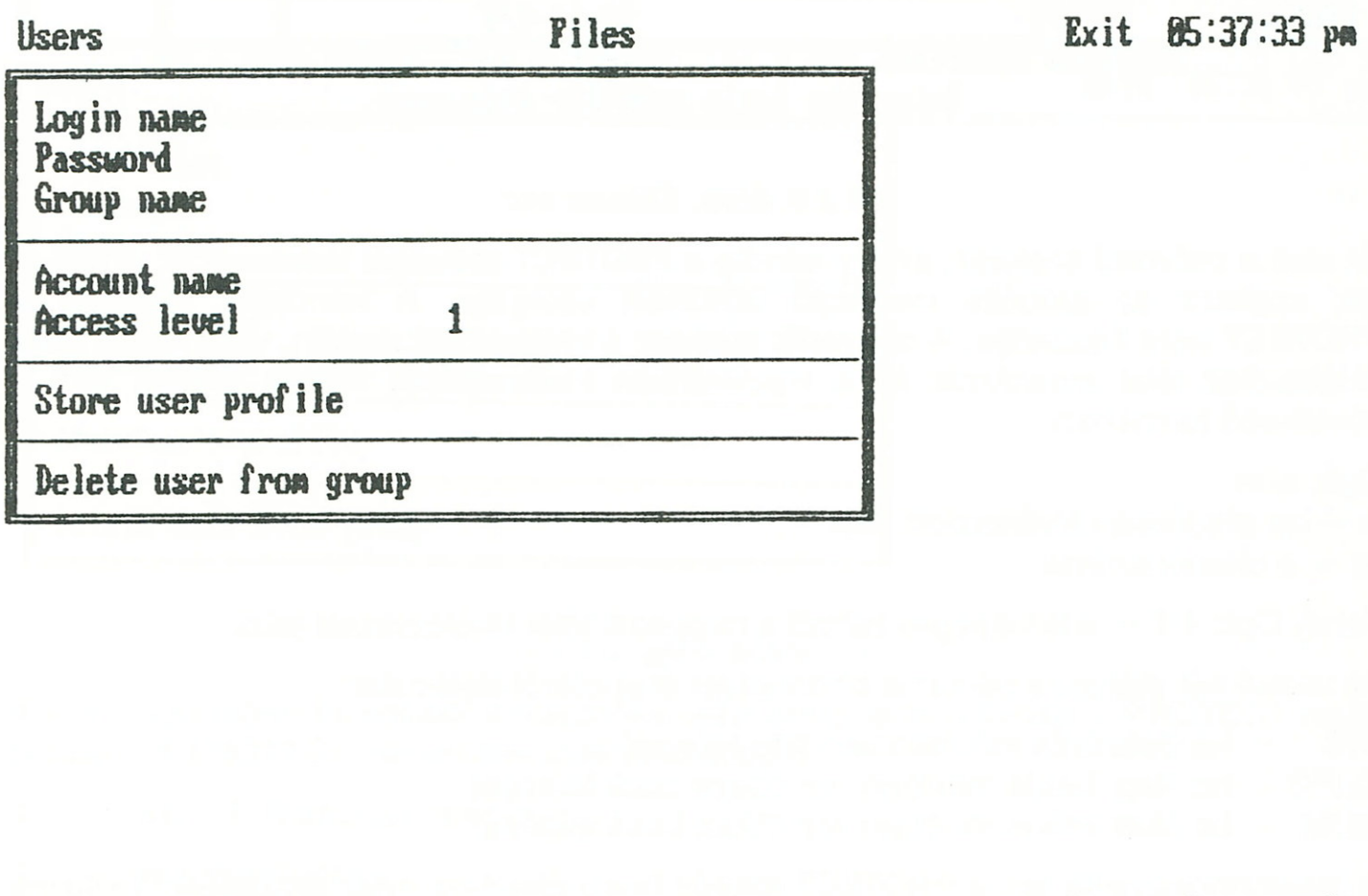
A huszonötödik, egyben utolsó sor az aktuális menü tétel használatával, bevitelével kapcsolatos információkat szolgáltatja.

13.3.3.2. A PROTECT használata

13.3.3.2.1. Profil Létrehozása

A felhasználói profil a felhasználó hálózati jellemzőinek gyűjteménye. A felhasználói (Users) menü hívatott a felhasználói profilok létrehozására, változtatására és törlésére.

A DBSYSTEM.DB titkosított file tárolja a felhasználói profilokat. Az alábbi ábra a Felhasználói (Users) menü szerkezetét illusztrálja.



13.3.6. ábra. Users menü (Felhasználói menü)

Az egyes tételek a felhasználói profil elemei:

*** Login name — A bejelentkezési név**
8 karakteres, az adatadminisztrátor által meghatározandó adat

*** Password — jelszó**
16 karakteres, az adatadminisztrátor által meghatározandó adat
A maximális 16 karakter használata ajánlott.

*** Group name — Csoport név**
8 karakteres, az adatadminisztrátor által meghatározandó adat

A csoport neve a file kódolás alapja. A csoport név alapján történik a dBASE III PLUS file kódolása. A file dekódolása csak a felhasználói és a file csoport azonossága esetén lehetséges.

Ajánlatos a felhasználókat az egyes alkalmazásoknak megfelelően csoportokba osztani. Az egyes alkalmazásokban szereplő felhasználók általában ugyanahhoz a felhasználói csoporthoz tartoznak.

*** Account name – azonosító**
25 karakteres, az adatadminisztrátor által meghatározandó adat

A felhasználói profil egyetlen opcionális eleme. Bármire használható, ajánlatos azonban a felhasználói profil egyértelműbb meghatározására használni. Maga a rendszer semmire sem használja.

*** Access level – Hozzáférési szint**
1-8 közötti egész szám, a felhasználók kategóriákba sorolásának, a file és mező előjogokkal együtt a file hozzáférések szabályozásának eszköze.

Minden csoporton belül a felhasználókhoz adott hozzáférési szintet rendelünk (más szóval a felhasználókat adott kategóriákba osztjuk). A felhasználói kategóriák és a Files menüben meghatározható file előjogszintek viszonya határozza meg a felhasználó file hozzáférési lehetőségeket, vagy teszi a file-t hozzáférhetetlenné a felhasználó számára. Alapértelemzés szerint a hozzáférési szint 1, megváltoztatása vagy megfelelő érték bevitelével vagy a kurzort le-fel mozgató billentyűk segítségével érhető el.

*** Store user profile – Felhasználói profil tárolása**

A felhasználói profil meghatározását befejezve, kijelöljük ez utóbbit DBSYSTEM.DB file-ba történő felvételre. A tulajdonképpeni profil tárolás az Exit menü Save vagy Exit pontjának kiválasztásakor valósul meg.

A felhasználói profil változatására és törlésére akkor nyílik lehetőségünk, ha a Felhasználói (Users) menü első három tételét helyesen adtuk meg . Ebben az esetben a többi tétel a megfelelő, létrehozáskor vagy előző változtatáskor meghatározott formájában jelenik meg. Ezek után bármelyik tétel megváltoztatható, de a már elmondottak alapján a csoport név (Group name) megváltoztatása nem ajánlott.

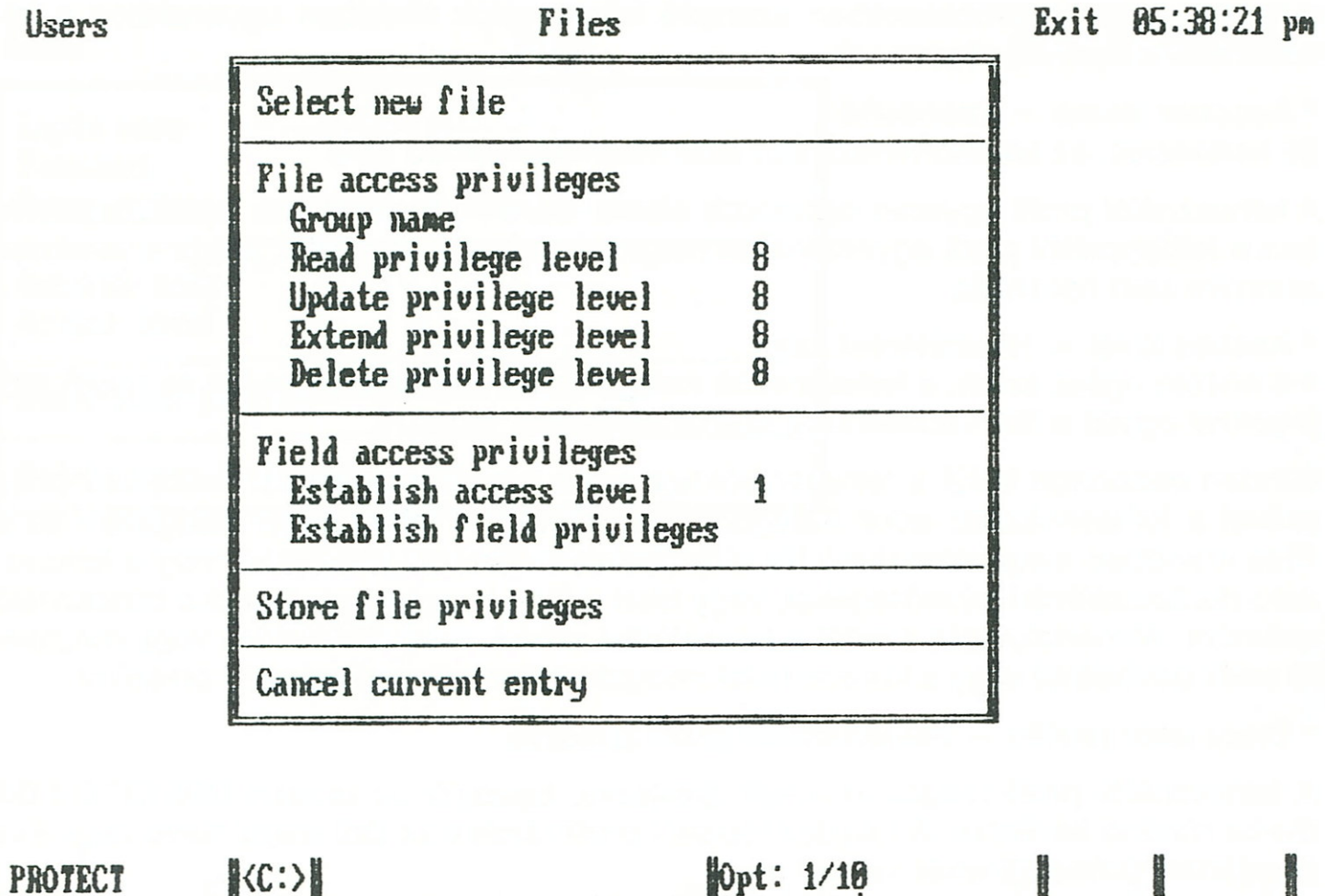
*** Delete user from group – Felhasználó törlése a csoportból**

A Users menü utolsó tétele lehetőséget kínál a profil törlésére. Akárcsak a változtatásnál, itt is a bejelenkezési név, jelszó és csoport helyes megadására van szükség. A "Delete user from group" tételre mozgatva az inverz mezőt, majd Enter-t leütve a felhasználói profil törlődik.

13.3.3.2.2. File és mező hozzáférési előjogszintek meghatározása

A Files menü feladata a file és mező hozzáférési előjogok szintjének beállítása.

A file privilégiumok elmentése az adatbázis file szerkezetébe történik, tehát a DBSYSTEM.DB változtatása nem érinti a file hozzáférési előjogok révén megvalósított védelmét.



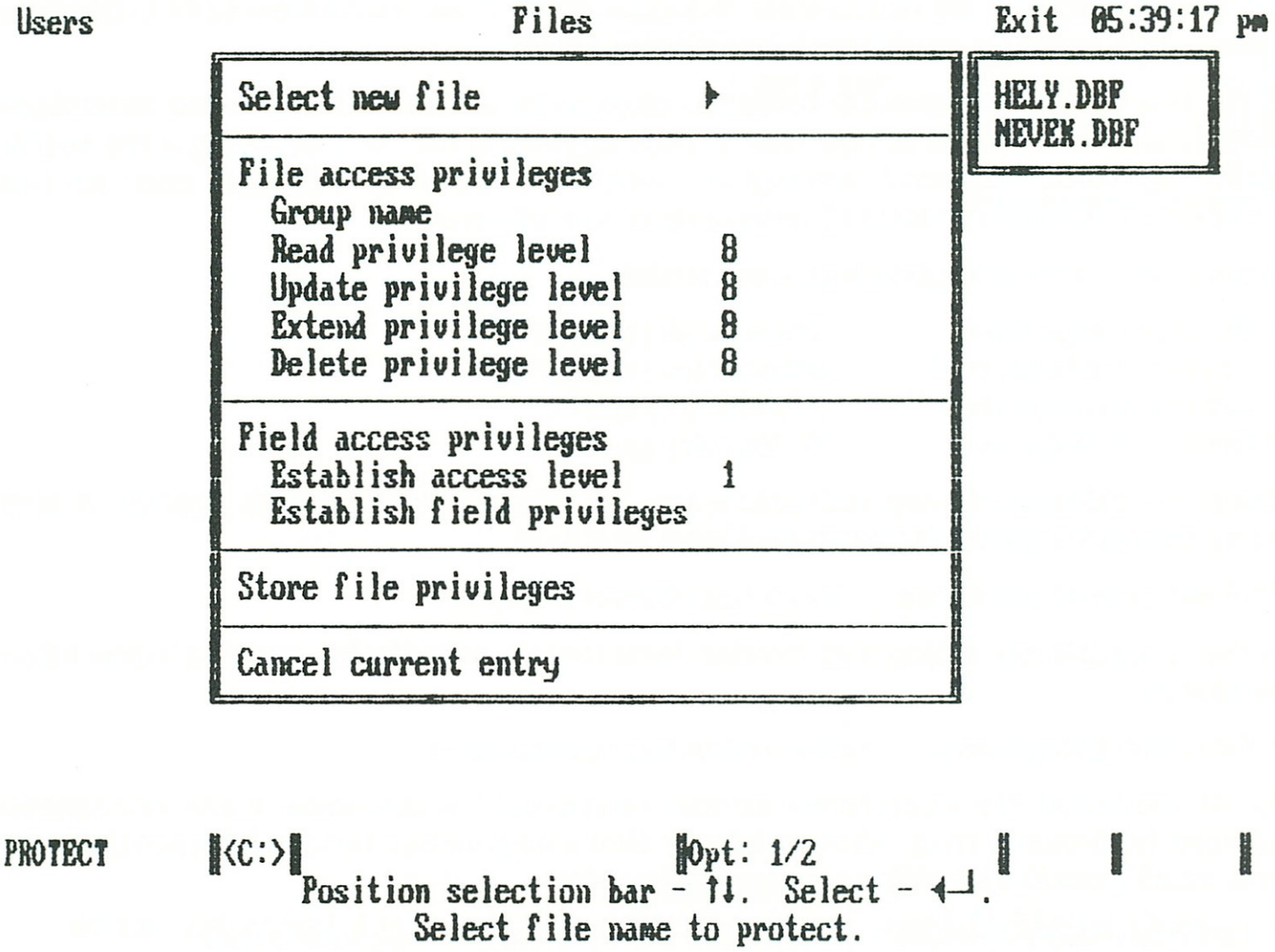
13.3.7. ábra. Files (File menü)

Az egyes menütételek:

*** Select new file — Új file kiválasztása**

A menütétel kiválasztása, majd az Enter leütése nyomán, a rendszer adatbázis file-jainak listája jelenik meg, a megfelelő file kiválasztása ugyancsak egy inverz kijelzésű mező mozgatásával érhető el. A visszatérés a file-ok listájából a Files menübe a kurzort balra mozgató nyíl segítségével valósítható meg.

Egyszerre legfeljebb 10 adatbázis file előjogrendszere hozható létre. Utána feltétlenül a Exit menü, Save (Mentés) vagy Exit (Kilépés) pontjának kell következnie,csak ezután folytatható további file-ok előjogrendszereinek meghatározása.



13.3.8. ábra. File kiválasztása

*** Group name – Csoport neve**
8 karakteres, üzemeltető által meghatározandó adat

A File csoport és a Felhasználói csoport neve (User group name), ha azonosak, a file titkosítás automatikusan feloldódik a felhasználói állomáson. A file-ok és felhasználók csoportokba szervezése ezért a lehetőségek szerint igazodjon az alkalmazói program teremtette körűlményekhez.

**** File access privileges – File hozzáférési előjogszintek**
1-8 közötti egész számok

Következik a file előjogszintek beállítása, amelyek megszabják file-hozzáférés feltételeit. A file előjogszintek 1-8 közötti számmal adhatók meg, az adott előjogszintnél kisebb vagy egyenlő hozzáférési szinttel (Access level) rendelkező felhasználó az adott előjoghoz hozzájut.

Ebből következik, hogy az 1-es hozzáférési szinttel rendelkező felhasználó rendelkezik a lehető legtöbb file hozzáférési előjoggal, mig a 8-as hozzáférési szint biztosítja a felhasználó számára az előjogok legszűkebb körét.

A file hozzáférést legkevésbé korlátozó előjogszint a 8-as hiszen minden lehetséges felhasználó hozzáférési szintje kisebb vagy egyenlő 8-nál. Az 1-es pedig a file hozzáférést leginkább korlátozó privilégium szint, mivel ezzel az előjoggal csak az 1-es hozzáférési szinttel rendelkező felhasználók rendelkeznek.

A file hozzáférési előjogbeállítás menütételei:

* Read privilege level — Olvasási előjogszint
* Update privilege level — Módosítási előjogszint
* Extend privilege level — Bővítési előjogszint
* Delete privilege level — Törlési előjogszint

Adott file előjogszint meghatározása egy 1-8 közötti szám bevitelét igényli. A fenti négy előjogot egymástól függetlenül lehet beállítani.

** Field access privileges — Mező hozzáférési előjogok

A mezőhozzáférési előjogokat minden felhasználói hozzáférési szinthez külön-külön rendeljük.

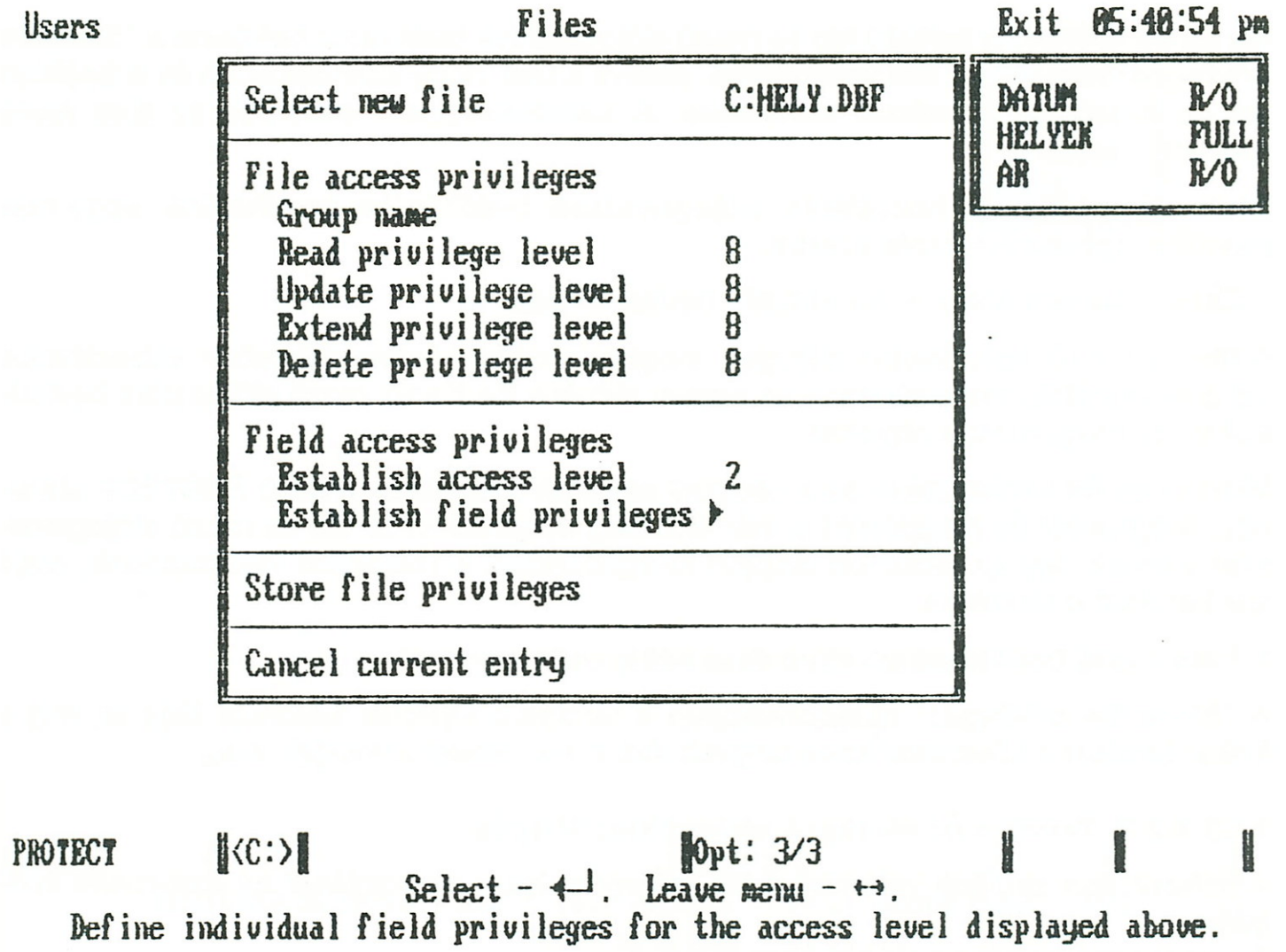
* Establish access level — Felhasználói hozzáférési szint

Az itt megadott file hozzáférési szinttel rendelkező felhasználók mező hozzáférési szintjeit határozzuk meg. Azonos felhasználói kategóriához tartozó felhasználók azonos mező hozzáférési előjogokkal rendelkeznek.

* Establish field privileges — Mező hozzáférési előjogok meghatározása

A menütétel kiválasztása az adott file mezőinek megjelenését eredményezi, a hozzájuk rendelt előjogokkal együtt. A mező előjogok lehetséges szintjei:

FULL — Olvasható és írható a mező
R/O — Csak olvasható a mező
NONE — A mező se nem írható, se nem olvasható



13.3.9. ábra. Files menü és mezők listája

Alapállapotban minden mezőhöz a FULL előjog tartozik. A megfelelő mező kiválasztása a mezők listájában az inverz kijelzésű mező le-fel mozgatásával történik, a megfelelő irányba mozgató billentyűkkel. Az előjog változtatás a FULL, R/O, NONE tartományban az Enter ismételt leütésével érhető el. A megfelelő előjog beállítása után továbblépheténk a következő mezőre.

A Files menübe a kurzort balra mozgató billentyű leütésével térhetünk vissza.

Ujabb felhasználói hozzáférési szint beállításával (Establish access level) egy másik felhasználói kategória mezőhozzáférési előjogait határozhatjuk meg az előbbiekben vázolt módon.

Megjegyzés: A file előjogok szélesebb körű korlátozást képviselnek a mező privilégiumoknál. Éppen ezért egy adott felhasználó által csak olvasható file esetében csak két értelmes mezőprivilégium létezhet a R/O és a NONE.

* Store file privileges — A file és mező privilégiumok tárolása

Ha az adott file-hoz tartozó file és mező előjogszintek beállítását befejezte a "Store file privileges" menütételt kell választania, jelezve ezzel, hogy szándékában áll a beállított file és mezőelőjogrendszer elmentése. A tulajdonképpeni elmentés az Exit menü feladata.

Ezek után újabb file hozzáférési előjogrendszer beállításába kezdhetünk, vagy más menübe léphetünk tetszés szerint.

* Cancel current entry — Az aktuális bevitel törlése

A file és mező hozzáférési előjogok meghatározása közben bármikor választhatjuk ezt a menütételt, megszüntetve az éppen aktuális file file és mező előjogszint beállításokat, de megőrizve a régieket.

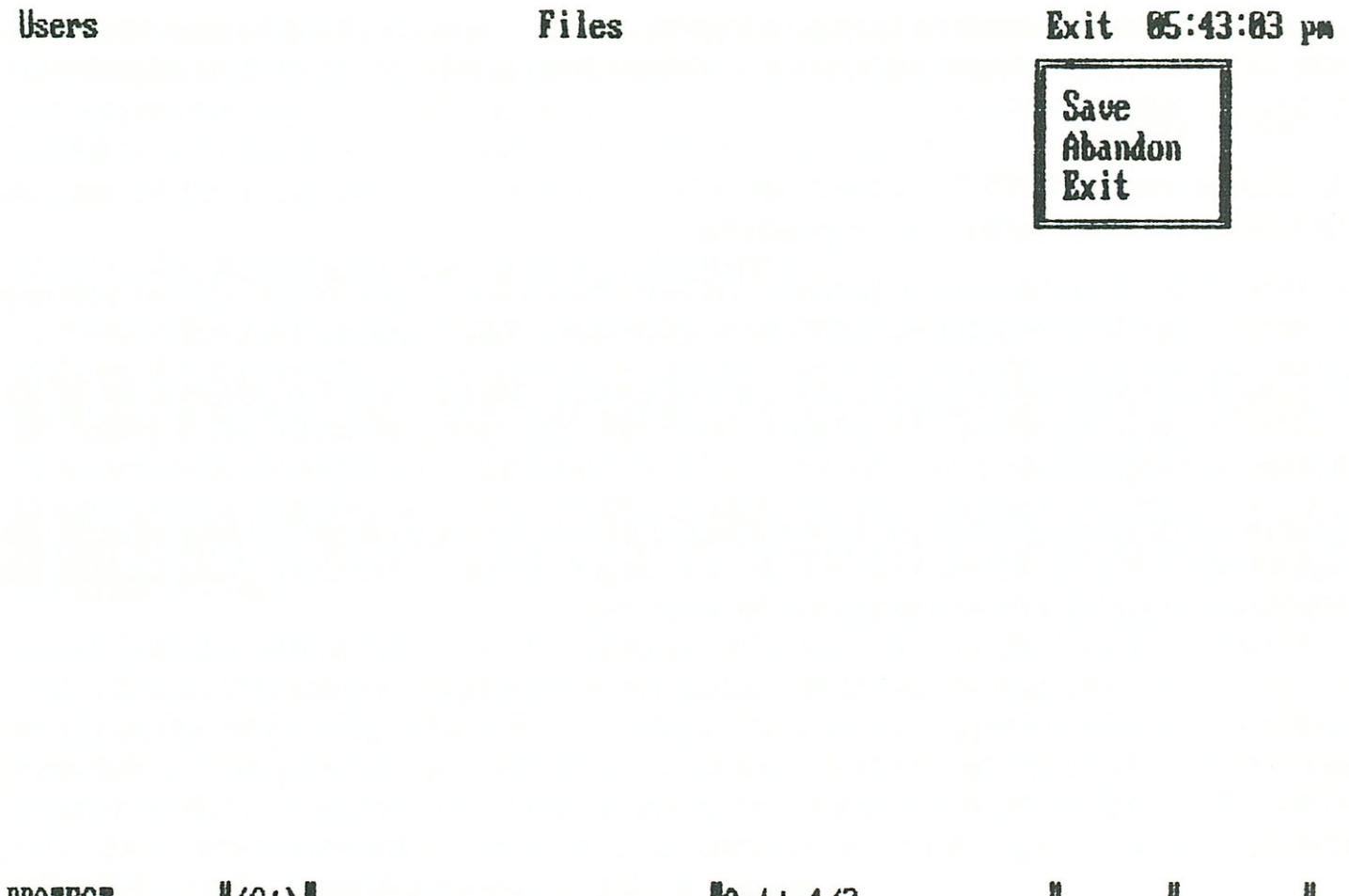
Minden új file kiválasztása és a csoport nevének megadása után a PROTECT utánanéz, megkeresi és megjeleníti a már előzőleg meghatározott file és mező előjogrendszer elemeit. Igy az előbbiek alapján elvégezhetőek a megfelelő változtatások, majd sor kerülhet a tárolásra.

A Files menü beállításainak elmentése kétlépcsős folyamat:

A "Store file privileges" tulajdonképpen a tárolásra kijelölés feladatát látja el, míg a fizikai tárolást a következőkben tárgyalt Exit menü tételei valósitják meg.

13.3.3.2.3. Profilok és előjogok elmentése; Kilépés

A felhasználói profilok valamint a file előjogrendszer elementését az Exit menü szolgálja.
A menü szerkezetét az alábbi ábra illusztrálja.



13.3.10. ábra. Exit menü

A menütételek:

*** Save — Mentés**

Az aktuálisan létrehozott vagy változtatott és tárolásra kijelölt felhasználói profilok és file/mező előjogszintrendszer elmentését valósítja meg. A felhasználói profilok, mint azt az előbbiekben vázoltuk, a DBSYSTEM.DB file-ba kerülnek felvételre,mig a file és mező előjogszintek maguknak az adatbázis file-oknak a struktúrájába épülnek be.

Minden tíz tárolásra kijelölt file és mező előjogot el kell menteni. Az adatbázis file-ok titkosított változata éppen a privilégium halmazok elmentésével egyidőben jön létre, egy .CRP kiterjesztésű file formájában.

A PROTECT befejezése után mentsük el, majd töröljük a nem titkosított .DBF file-t, majd a .CRP kiterjesztésű titkosított változatot nevezzük át .DBF kiterjesztésűvé.

*** Abandon – A még el nem mentett profilok és privilégiumok törlése**

Az aktuálisan létrehozott és tárolásra kijelölt, de még nem elmentett felhasználói profilok, valamint file és mező előjogszintek törlését lehet e menüfunkció révén elérni.

*** Exit – Kilépés**

Az Exit jelenti a PROTECT befejezését. Kilépés előtt a már előbb vázolt SAVE menütétel funkcióját automatikusan megvalósítja.

A PROTECT használatának jogával nyilvánvalóan csak korlátozott számú személy rendelkezhet. Üzemeltetése a hálózati Adatadminisztrátor joga és feladata.

A PROTECT jelszóvédett program, használatának elengedhetetlen feltétele az adminisztrátor jelszóismerete. Az adatadminisztrátornak gondoskodnia kell a jelszó biztonságos megőrzéséről.

Adatvédelmi szempontból az adatadminisztrátor különleges jogokkal rendelkezik, de ugyanakkor fontos követelményeknek kell eleget tennie, a hálózati alkalmazásoknak megfelelő adatvédelmi stratégiát kell kidolgoznia.

13.4. Programozás hálózati környezetben

A hálózati környezet mint többfelhasználós rendszer újabb feladatok elé állítja a programozót. A dBASE III PLUS hálózati eszköztára megnyugtató megoldást kínál minden egyes kérdésre. A hálózati rendszer működtetésének kritikus pontjai az osztott adatbázis kezelés egyenes következményeként jelentkeznek.

13.4.1. Az adatsérthetetlenség védelme

Az adatbázis file-ok megosztott használata által felvetett leggyakoribb probléma az ütközés. Ez a jelenség akkor keletkezik, amikor több felhasználó próbálja egyszerre ugyanazt az adatot módosítani vagy egyszerre akar új adatot fűzni az adatbázishoz.

De szemléltessük rögtön egy példával, mi is történik ilyenkor. Tekintsünk egy egyszerű színházjegy eladási rendszert. Több jegypénztár számítógépe kapcsolódik hálózatban egymáshoz és egy, a műsornaptár bővítési és törlési jogaival egyedül felruházott főpénztárhoz.

Ha az 1-es pénztár a 88/01/02-én esedékes előadás még szabad helyeiből még hármat szándékozik eladni, megkeresi a megfelelő dátumhoz rendelt rekordot SEEK-kel és azt találja hogy még HELYEK = 10 szabad hely akad. Ugyanezt találja egy másik, mondjuk a 2-es pénztár is, tehát itt is az olvasás eredménye HELYEK = 10. Az 1-es pénztár eladja a 3 jegyet, a szabad helyek száma most 7 a REPLACE HELYEK WITH HELYEK-3 eredményeként. Most a 2-es pénztár ad el 2 jegyet, az adatbázisba HELYEK = 10-2 = 8 kerül a helyes 10-3-2 = 5 helyett.

```
      1-es pénztár              2-es pénztár
      ------------              ------------

SEEK "88/01/02"               SEEK "88/01/02"
@ 10,10 SAY HELYEK            @ 10,10 SAY HELYEK
@ 12,10 GET t_elad            @ 12,10 GET t_elad
READ                          READ
-----------------------------------------------
REPLACE HELYEK WITH HELYEK-t_elad
                         majd
                              REPLACE HELYEK WITH HELYEK-t_elad
```

Az ütközés miatt csak egy felhasználó változtatási parancsa lesz eredményes, a többié nem, miközben meggyőződésük, hogy ők is megváltoztatták az adott rekordot.

Minden hálózati adatbázis rendszernek védelmet kell nyújtania az egyidejű adathozzáféréssel szemben, vagyis elejét kell vennie az olyan helyzeteknek, amikor két felhasználónak egyidejűleg saját másolata van ugyanarról a rekordról, miközben mindkettő változtatásra készül. A megoldás a zárolás technikájában rejlik.

A DBASE III PLUS két zárolási lehetőséget ismer:

- file zárolás
- rekord zárolás

Rekord zárolás beépítésével az előbbi példa a következőképpen fest:

```
     1-es pénztár              2-es pénztár
     ------------              ------------

SEEK "88/01/02"
* zárolási kisérlet
* zárolás sikeres
* következhet a
* változtatás                 SEEK "88/01/02"
                              @ 10,10 SAY HELYEK
@ 12,10 GET t_elad            * ismételt zárolási kisérlet
                              * zárolás nem engedélyezett
READ                          * várakozás
REPLACE HELYEK WITH HELYEK-t_elad
* zárolás feloldása
                              * sikeres zárolási kisérlet
                              @ 10,10 SAY HELYEK
                              @ 12,10 GET t_elad
                              READ
                              REPLACE HELYEK WITH HELYEK-t_elad
                              * zárolás feloldása
```

Az egyik felhasználó sikeres zárolása, a másik felhasználó zárolási kisérleteit meghiúsítva ez utóbbit várakozó álláspontra kényszeríti a zárolás feloldásáig.

A dBASE III PLUS a zároláson kívül a megnyitás szabályozásával is elejét veheti a nem kívánt konkurrens, egyidejű adathozzáférésnek.

A file-okat kétféleképpen nyithatjuk meg:

- kizárólagos (exclusive) módon
- megosztott (shared) módon

A kizárólagosan megnyitott file, csak egy felhasználó számára hozzáférhető, tehát ilyen esetben fölösleges további zárolások használata. Kizárólagosan nyitjuk meg általában az adott felhasználó egyedi, saját file-jait.

A megosztott módon megnyitott file több felhasználó számára hozzáférhető, s így a konkurrens használat lehetőségét biztosítja. Az adatváltoztatás ilyenkor megkövetelheti a file vagy rekord zárolását.

13.4.1.1. Zárolás a dBASE III PLUS-ban

A zárolásra a file-ok megosztott használáta esetén van szükségünk. A zárolásnak három szintje ismert a DBASE III PLUS-ban. Ezek precedenciájuk csökkenő sorrendjében a következők:

1. Automatikus file zárolás. A DBA - dBASE ADMINISTRATOR üzemeltetőjétől független zárolás (beépített DBA funkció). Mindazok a parancsok amelyek az egész file-t igénybe veszik, csak a file zárolt állapotában használhatók. A nem kizárólagosan használt file-ok automatikus zárolásáról minden előbb említett esetben a DBASE gondoskodik.

A következő parancsok előtt a dBASE automatikusan zárolja az adott file-t:

APPEND, APPEND FROM, AVERAGE, BROWSE, COPY, COPY STRUCTURE, COUNT, DELETE ALL, INDEX, JOIN, RECALL ALL, REPLACE ALL, SORT, SUM, TOTAL, UPDATE

2. Explicit file-zárolás. A programozó dBASE parancsok révén maga is zárolhat file-okat, és a file-ok zárolását feloldhatja. A file zárolás során a felhasználó, illetve alkalmazói program a file módosítások idejére zárolja az egész file-t, majd a változtatások befejeztével feloldja a file zárolását.

   Az explicit file-zárolás különösen jó szolgálatot tehet ha a módosítások sok rekordot illetve index file-t érintenek. A file zárolása mellett nyilvánvalóan nincs szükség rekord zárolásra.

3. Explicit rekord-zárolás. A file-on belül egy rekord zárolása csak egyedül ennek a rekordnak a hozzáférését teszi lehetetlenné a többi felhasználó számára. A rekord zárolás kizárja ugyanannak a rekordnak több felhasználó általi egyidejű módosításának lehetőségét.

   A rekord-zárolás a következő lépésekben történik:
   - Rekord-zárolás
   - A változtatandó adat beolvasása és vizsgálata. Mindig az aktuális adat jelenik meg. Ezt az adatot más nem változtathatja.
   - A módosítás végrehajtása és a módosított rekord tárolása
   - A rekord-zárolás feloldása

A fejezet elején vázolt zárolási példában az események fonala ezeknek a lépéseknek megfelelő.

13.4.1.2. A dBASE III PLUS zárolási eszköztára

13.4.1.2.1. Teljes képernyős (full screen) rekord megjelenítési, módosítási parancsok és a rekord-zárolás.

A dBASE ADMINISTRATOR (DBA) a teljes képernyős megjelenítés és módosítás lehetőséget kínáló (full screen) EDIT és CHANGE parancsai esetén egyszerű billentyűkombináció leütésével is elérhető az adott rekord zárolása, illetve a rekord zárolásának feloldása. A rekord szerkesztés kétlépcsős parancs, a rekord megjelenítését és a változtatás lehetőségét foglalja magába. A rekord megjelenítésénél nem követelmény a zárolás, annál inkább az a rekord változtatásnál.

A megjelenített rekord zárolása a CTRL-O billentyűk lenyomásával érhető el. A zárolás feloldását újabb CTRL-O, vagy CTRL-W leütésével érhetjük el. A rekord szerkesztés befejezésével egyébként a rekord zárolása automatikusan feloldódik. Az EDIT és CHANGE full-screen parancsok státusz sorának hálózati működtetés számára fentartott részében mindig megjelenik a rekord zárolási állapota.

A rekord zárolása előtt, az EDIT és CHANGE parancsok esetén az adatváltoztatás lehetetlen. A zárolás alapvető követelmény e parancsok hálózati használatánál.

13.4.1.2.2. File/Rekord zárolási parancsok

A zárolás parancsai minden más DBASE III PLUS parancstól eltérő tulajdonságokkal rendelkeznek. Sajátos szerkezetük és működésük alapján függvénybe "torkolló" parancsoknak is tekinthetjük ezeket. Mint parancs a zárolás működésbe hozza a dBASE zárolási mechanizmusát, a dBASE pedig a maga részéről továbbítja a zárolási kérelmet a hálózati operációs rendszer szintjére. A fizikai zárolást a hálózati operációs rendszer kísérli meg. Itt "fordul" a parancs függvénybe. Ha a zárolás sikeres a függvényérték .T. (true = igaz) ellenkező esetben .F. (false = hamis)

A zárolás tehát egy parancs, amely zárolja a file-t/rekordot és egy függvény, amely a zárolási kísérlet eredményéről számol be. Csak nem zárolt állapotú file/rekord zárolható sikeresen.

- A file zárolási parancs: FLOCK()
- A rekord zárolási parancs: RLOCK() vagy LOCK()
- A file/rekord zárolás feloldási parancsa: UNLOCK

A zárolási parancsok ellenörzik a file/rekord zárolási állapotát, majd ha file/rekord nem zárolt, zárolják és .T. értékkel térnek vissza. Ha file zárolt, minden újabb zárolási kísérlet sikertelen, és a parancs/függvény .F. értékkel tér vissza.

A zárolás feloldását vagy az UNLOCK paranccsal, vagy egy másik rekord zárolásával, ugyanazon a munkaterületen új file zárolásával vagy file lezáró parancsokkal érhetjük el. Ez utóbb emlitett parancsokhoz tartozik a CLOSE, USE, CLEAR ALL és a QUIT.

13.4.1.2.3. File-zárolás. Követelmény és ajánlat

Adatbázis file-ok zárolása esetén a file rekordjai minden más felhasználó számára hozzáférhetetlenek. A file zárolás sokkal több ügylet (tranzakció) konfliktust okoz, több felhasználónak kell várnia a zárolás feloldására. A file-zárolás alacsonyabb hálózati teljesítőképességű, mint a rekord-zárolás.

Egyes műveletek megkövetelik a file zárolását, és ha ez nem történt meg, a dBASE automatikusan zárol. Az automatikus file-zárolást előidéző parancsokat fentebb soroltuk fel. A file-zárolás igazi haszna akkor jelentkezik, mikor több rekord egyidejű változtatására van szükség, mert fölöslegessé teszi a többszörös rekord zárolást, ami ilyen esetben időigényesebb lenne.

13.4.1.2.4. Rekord-zárolás. Követelmény és ajánlat

Ha több felhasználó használja ugyanazt a file-t, a rekordzárolás ajánlott, hiszen egy rekord zárolása után a többi rekord hozzáférhető marad. A @ sor,oszlop GET "mezo" READ, REPLACE és DELETE parancsok megkövetelik a rekord zárolást.

Mindhárom parancs egyszerre egy adott rekordra vonatkozó műveletet, változtatást feltételez. Változtatás csak zárolt rekordokon célszerű végrehajtani.

13.4.1.2.5. File/rekord-zárolás időtartama

Célszerű a zárolási időtartam korlátozására törekednünk, ez világos, hiszen a zárolt file-hoz, rekordhoz a zárolás ideje alatt senki más nem férhet hozzá.

A zárolás idejének csökkentése érdekében a következő tényezőket kell figyelembe vennünk:

- a zárolás típusa
- az FLOCK() vagy RLOCK() és UNLOCK közötti parancsok száma és típusa

A zárolás és a zárolás feloldása közötti

@ sor,oszlop GET "változó"
READ

parancssorozat képes meghatározatlan időre megőrizni a file/rekord zárolt állapotát. Felfüggeszti a program futását és adatbevitelt vár a képernyőről. Ha az adatbevitel bármilyen okból is – és ilyen okok könnyen születnek – késik, ez meghosszabítja a file zárolási idejét, több felhasználót várakozásra kényszerít, s így komoly konfliktushelyzeteket is teremthet. A zárolási idő lerövidítése érdekében az adatbeviteli eljárások (READ) használatát szigorú szabályokhoz ajánlott kötni.

13.4.1.2.6. Több file zárolása

Több file zárolását az egyes munkaterületek kiválasztása, majd az FLOCK() parancs kiadása valósítja meg.

Az EGY és KETTO munkaterületeken két file-t nyitottunk meg.

```
SELECT EGY
DO FZAR
SELECT KETTO
DO  FZAR
```

```
* FZAR
ismet=.t.
DO WHILE .NOT. FLOCK().AND.ismet
  @  23,20 SAY " A  File-t más már;
  használja !! Még próbalkozik?(i,n)";
  GET ismet PICTURE "@i"
  READ
  IF .not.ismet
     @ 23,36 say " Müvelet vége"
     UNLOCK ALL
     CLEAR ALL
     RETURN
  ENDIF
ENDDO
```

13.4.1.2.7. Több rekord zárolása

Több rekord zárolása két úton érhető el:

Az előbbi példához hasonlóan munkaterületek váltogatásával és RLOCK() kiadásával.

```
SELECT EGY              * RZAR
DO RZAR                 ismet=.t.
SELECT KETTO            DO WHILE .NOT. RLOCK().AND.ismet
DO  RZAR                  @  23,20 SAY "A Rekordot más már;
                          használja !! Még próbalkozik?(i,n)";
                          GET ismet PICTURE "@i"
                          READ
                          IF .not.ismet
                             @ 23,36 say " Művelet vége"
                             UNLOCK ALL
                             CLEAR ALL
                             RETURN
                          ENDIF
                        ENDDO
```

Az ún. "view-lock" segitségével; a "view" a elnevezés VIEW (.VUE) file-ok nevéből származik. Tulajdonképpen egy "view"-t adunk meg, lásd (VIEW file-ok), majd a mezők beállítása (SET FIELDS ON) utan, kiadjuk az RLOCK() parancsot.

```
SELECT EGY
USE file1 INDEX ifile1
SELECT KETTO
USE file2 INDEX ifile2
SET RELATION TO kulcs2 INTO EGY
SET   FIELDS   TO  egy   →    kulcs1,;
egy →   szoveg1,kulcs2, szoveg2
SET FIELDS ON
RLOCK()
Edit
```

"kulcs1" a file1 indexelésének kulcsa.

"kulcs2" a file2 indexelésének kulcsa.

Az RLOCK() a "view"-t felepítő összes rekordot zárolja. Ezután lehetőség nyílik egy "logikai rekord" változtatására, amely a két file különböző mezőiből épül fel.

13.4.1.3. A file megnyitás és a hozzáférés

A file megnyitás két módját ismeri a DBASE III PLUS:

Exclusive - Kizárólagos megnyitás

Az így megnyitott file csak egy felhasználó számára lesz hozzáférhető. A file kizárólagos megnyitása fölöslegessé teszi a file/rekord-zárolást.

Shared - Megosztott módban történő megnyitás

Az így megnyitott file több felhasználó számára hozzáférhető. A file/rekord-zárolást csak a megosztott módban megnyitott file-ok igénylik.

A dBASE segítségével a file megnyitás módja megfelelően karbantartható. A karbantartást az EXCLUSIVE kapcsoló megfelelő állításával végezhetjük.

SET EXCLUSIVE ON parancs után megnyitott file-ok a kizárólagos mód szabályai szerint hozzáférhetők.

SET EXCLUSIVE OFF kikapcsolja az alapértelmezés szerinti kizárólagos megnyitási módot. A parancs után "egyszerű" USE-zal megnyitott file-ok megoszthatók, tehát több felhasználó használhatja őket. A SET EXCLUSIVE OFF után bármely file megnyitható kizárólagos módban is, a következő USE paranccsal:

USE file_név INDEX index_név EXCLUSIVE ALIAS alias_név

Az előbbiekben az "egyszerű" USE megnevezést az EXCLUSIVE nélküli USE parancs megjelölésére használtuk.

A file megnyitás után kiadott SET EXCLUSIVE parancsok a file hozzáférhetőségét már semmiképpen sem befolyásolják.

A dBASE ADMINISTRATOR (DBA) alapértelemezés szerint az adatbázis file-okat kizárólagos módban nyitja meg, tehát a SET EXCLUSIVE kapcsoló "ON" beállítása van érvényben.

A file-típusok alapértelemzés szerinti megnyitási módjait az alábbi táblázat összegzi:

File típus	Megnyitás módja	File megnyitó Parancs
ADATBÁZIS (.DBF)	Kizárólagos	CREATE
	Kizárólagos	USE
	Kizárólagos	COPY STRUCTURE TO
	Kizárólagos	COPY TO
	Kizárólagos	JOIN TO
	Kizárólagos	TOTAL TO
	Megosztott	APPEND FROM
	Kizárólagos	CREATE ba_file
	Megosztott	FROM bol_file
	Megosztott	UPDATE FROM
ALTERNATE (.TXT)	Kizárólagos	SET ALTERNATE TO
CATALOG (.CAT)	Megosztott	SET CATALOG TO
PROGRAM (.PRG)	Kizárólagos	MODIFY COMMAND
	Megosztott	DO
FORMAT (.FMT)	Megosztott	SET FORMAT TO
	Kizárólagos	MODIFY COMMAND
INDEX (.NDX)	Kizárólagos	INDEX
	Kizárólagos	USE
	Kizárólagos	SET INDEX TO
MEMO (.DBT)	Kizárólagos	
LABEL (.LBL)	Kizárólagos	CREATE/MODIFY LABEL
	Megosztott	LABEL FROM
MEMORY (.MEM)	Kizárólagos	SAVE

	Megosztott	RESTORE
PROCEDURE	Kizárólagos	MODIFY COMMAND
	Megosztott	SET PROCEDURE TO
QUERY (.QRY)	Kizárólagos	CREATE/MODIFY QUERY
	Megosztott	QUERY
REPORT (.FRM)	Kizárólagos	CREATE/MODIFY REPORT
	Megosztott	REPORT FROM
VIEW (.VUE)	Kizárólagos	CREATE/MODIFY VIEW
	Megosztott	SET VIEW TO

Az index és memo file-ok megnyitási módja mindig azonos a velük kapcsolatban álló adatbázis file-ok megnyitási módjával. A SET EXCLUSIVE parancs csak az adatbázis file-ok és a hozzájuk tartozó index és memo file-ok megnyitási módját változtathatja.

Összegzésül elmondhatjuk: A CREATE/MODIFY műveletek mindig kizárólagos megnyitást eredményeznek, míg a file működésbe hozása (SET .. TO), aktiválása automatikusan megosztott file megnyitáshoz vezet.

Egyes parancsok a file kizárólagos megnyitását feltételezik, ellenkező esetben végrehajtásuk hibaüzenettel megszakad. Ezek:

INSERT, MODIFY STRUCTURE, PACK, REINDEX, ZAP

A file hozzáférhetőségéről a megnyitási módozatokon kívül, a PROTECT programmal, az operációs rendszer szintjén és hálózati szinten is rendelkezhetünk.

A PROTECT az adatbázis file-ok hozzáférhetőségét előjogszintekhez rendeli (lásd a PROTECT használatát).

Az operációs rendszer szintjén bármely file attributuma állítható az ATTRIB paranccsal. Adott file attributuma határozza meg hogy az adott file:

- írható/olvasható – R/W (read/write)

vagy

- csak olvasható – R/O (read only)

Írható/olvasható attributumú file tartalma olvasható és változtatható. Csak olvasható file tartalma, mint nevéből is következik, olvasható, a file változtatása viszont nem lehetséges (függetlenül a DBASE III PLUS használati módjától).

A hálózat szintjén a könyvtárhozzáférést is szabályozhatjuk, amint azt az adatvédelemnek szentelt fejezetben áttekintettük.

A dBASE ADMINISTRATOR az adatbázis file-okat írásra/olvasásra nyitja meg. A CREATE/MODIFY parancsok a megfelelő file-okat írásra/olvasásra nyitják meg.

A VIEW (.VUE), a QUERY (.QRY),a LABEL (.LBL), a FORMAT (.fmt) és a REPORT (.FRM) file-ok megnyitása csak olvasásra történik. (Kivétel a CREATE/MODIFY parancs.)

13.4.2. A holtpont és elkerülése

A holtpont, másnéven "halálos zárolás" a következő feltételek között léphet fel:

- több felhasználó próbál ugyanahhoz a file-hoz egyidőben hozzáférni sorozatos file kezelési eljárások végrehajtása céljából;
- mindegyik file-t más felhasználó nyitja, de nem ugyanaz zárja;
- egy zárolt file-t vagy rekord-ot újból zárolni akarunk;
- a program ciklikusan ismétlődik az eljárás (tranzakció) befejeztéig;

Az előbb vázolt esetekben végtelen ciklus az eredmény. Lássunk néhány példát:

ON ERROR DO HIBA
* A HIBA program ERROR() =108 esetén
* a file-t másvalaki használja RETRY
* parancsot ad, amely a HIBA befejezését
* és a HIBA bekövetkezte előtti parancs
* megismétlését eredményezi (lásd következő alfejezet)

1. példa

```
PR1:  ON ERROR DO HIBA          PR2: ON ERROR DO HIBA
      SET EXCLUSIVE ON               SET EXCLUSIVE ON
      SELECT 1                       SELECT 1
      USE FILE_1                     USE FILE_2
      SELECT 2                       SELECT 2
      USE FILE_2                     USE FILE_1
```

Az a helyzet állhat elő hogy a két program egymásra vár. A PR1 a PR2 által kizárólagosan megnyitott FILE_2, míg PR2 a PR1 által kizárólagosan megnyitott FILE_1 lezárását várja.

2. Példa

```
PR1:  SET EXCLUSIVE OFF          SET EXCLUSIVE OFF
      SELECT 1                   SELECT 1
      USE FILE_1                 USE FILE_1
      DO F_LOCK                  DO F_LOCK
      SELECT 2                   SELECT 2
      USE FILE_2                 USE FILE_2
      DO F_LOCK                  DO F_LOCK
```

Az F_LOCK eljárás a file zárolást ismétli amíg sikerül. Előfordulhat, hogy két program egymás zárolt file-ára vár.

3.Példa

```
PR1:  SET EXCLUSIVE OFF      PR2:  SET EXCLUSIVE OFF
      SELECT 1                     SELECT 1
      USE FILE_1                   USE FILE_2
                                   DO F_LOCK
      SEEK "F1"                    SELECT 2
```

```
DO R_LOCK                    USE FILE_1
SELECT 2          •          SEEK "F1"
USE FILE_2
DO F_LOCK
```

R_LOCK ismételten zárolni próbálja az adott rekordot. Előfordulhat, hogy PR1 a már zárolt FILE_2 -t a PR2 pedig a szintén zárolt "F1" kulcsú rekordra próbálja a rekordmutatót pozícionálni. Az ON ERROR DO HIBA program ismételten próbál a zárolt rekordra pozícionálni.

Az ilyen és ezekhez hasonló holtpontok elkerülése a programkészítő feladata. A halálos zárolás megelőzésének egyik módja az, hogy a nyitási és zárolási műveletek időrendi sorrendjét pontosan megtervezzük. A halálos zárolás programból történő megelőzése, mégis nehezebb feladat, mint annak észlelése és javítása. A file/rekord-zárolás és az ON ERROR parancs együttesen hivatott a végtelen ciklusok elkerülésére.

Az ON ERROR DO HIBAKEZELO parancs tulajdonképpen "monitor", megfigyelő szerepet játszik. A hiba észlelése nyomán indítja a HIBAKEZELO programot. Kérdező módszerrel egyik felhasználó megszakíthatja saját eljárását, s így lehetőséget biztosíthat a másik számára a folytatáshoz.

13.4.3. Hibák észlelése és az "újraélesztés"

A hibák észlelése és javítása különböző szinteken különbözőképpen történik.

- A pont kijelzés (prompt) szintjén a hibát a dBASE észleli, és hibaüzenettel tér vissza .
- Teljes képernyős megjelenítést előidéző (full-screen) parancsok esetén a hibaüzenet általában a képernyő 24. sorában, jelenik meg.
- Program file-ban a hibák kezelését az ON ERROR parancs látja el mégpedig a következőképpen:

  ON ERROR DO HIBA

- Hiba észlelése esetén a HIBA.PRG program indul, amely a hibákat kódjuk alapján azonosítja. Adott hiba kódjáról az ERROR() függvény segitségével szerezhetünk tudomást. A hiba ismeretében a HIBA program megfelelően járhat el az "újraélesztés" érdekében.
- Az ERROR() és MESSAGE() függvények csak akkor kapnak értéket, ha a programban létezik ON ERROR DO HIBA parancs.
- A hibakezelő programban jól hasznosítható parancs a RETRY. A RETRY az adott eljárás befejezését és a hívó programba való visszatérést eredményezi, mégpedig ugyanabba sorba ahonnan a hívás történt, ellentétben a RETURN paranccsal, amely programhívást követő sorba tér vissza.
- A hibakezelést a 13.4. fejezet végén szemléltetjük (HIBA.PRG).

13.4.4. Adatvédelem a programozás szintjén

Az adatvédelmet taglaló fejezetben ismerkedtünk az egyes felhasználókhoz és adatbázis file-okhoz rendelt hozzáférési szintekkel és ezek kapcsolatával.
Az ACCESS() függvény a felhasználóhoz rendelt hozzáférési szint értékét adja vissza. Ennek ismeretében a felhasználó programon belüli mozgástere is korlátozható.

Amint ez a fejezetet záró példaprogramból kitűnik, a 2 -es hozzáférési szinttel rendelkező pénztárak nem rendelkeznek az 1-es hozzáférési szinttel felruházott főpénztár műsornaptár bővítési és szűkítési (törlési) jogaival, így a példaprogram e két ága a pénztárak számára hozzáférhetetlen.

Íme ez a programszelet:

A LOGOUT parancs az adott felhasználó kijelentkezésének és egy másik felhasználó bejelentkezésének feladatát látja el. Segitségével a programozó ellenőrzést gyakorolhat a felhasználók ki-bejelentkezése fölött.

A LOGOUT hatására az állomás képernyője törlődik, és a már ismertetett (lásd adatvédelem) bejelentkezési képernyő jelenik meg. Érvényes csoportnév, felhasználónév és jelszó begépelése után, minden bizonnyal változnak a file-hozzáférési és programághozzáférési körülmények is.

Az ACCESS() függvény és a LOGOUT parancs használatának csak akkor van értelme, ha a rendszer a PROTECT védelme alatt áll.

13.4.5. Példaprogram

Egy jegyeladási rendszerről van szó. Adott műsornaptárba felvett előadások jegyeit forgalmazza a rendszer több pénztár és egy főpénztár révén. A főpénztár kitüntetett jogokkal rendelkezik. Egyaránt bővítheti és szűkítheti az előadások körét. Hozzá PROTECT-tel az 1-es hozzáférési szintet rendeltük. Az összes pénztár 2-es hozzáférési szinttel rendelkezik.

A HELY.DBF file szerkezete a következő:

Mező	Típus	Hossz
DATUM	Character	8
HELYEK	Numeric	2
AR	Numeric	3

A file írási, olvasási előjoga 2-es, a bővítési és törlési előjoga a PÉNZTÁR csoportban 1-es szinten helyezkedik el. Az 1-es hozzáférési szinttel rendelkező felhasználók a HELY mezőit illetően TELJES(FULL) jogkörrel rendelkeznek. A 2-es hozzáférési szint felhasználói csak a HELYEK mezőt illetően rendelkeznek TELJES(FULL) jogkörrel. A többi mezőt csak olvashatják, tehát mező előjoguk CSAK OLVASHATÓ (R/O).

File előjogok HELY.DBF

Előjogszint	1	2	3	4	5	6	7	8
Bővítés	*							
Törlés	*							
Olvasás		*						
Változtatás		*						

A "*" az előjogszint és a file előjog metszéspontjába kerül.

Mező előjogok

	Felhasználói hozzáférési szint							
Mező neve	1	2	3	4	5	6	7	8
DATUM	T	O	.	.	.	.	.	.
HELYEK	T	T	.	.	.	.	.	.
AR	T	O	.	.	.	.	.	.

"T" – TELJES (FULL) jogkör
"O" – CSAK OLVASHATÓ (R/O) jogkör

A HELY.DBF file DATUM mezője szerint indexelt. Az indexfile neve DHELY.NDX.

A program szerkezete:

JEGYEK.PRG - Főmenü

HIBA.PRG
Hibaészlelő és
újraélesztő program

HJ_FEL.PRG	HJ_TOR.PRG	HJ_ELAD.PRG	HJ_MEGJ.PRG
Műsornaptár bővítése	Törlés a műsornaptárból	Jegyeladás	Műsornaptár megjelenítése MUSOR.FRM szerint

A programlista:

```
*********************************************
*      Program : Jegyek.prg                 *
*                                           *
*      Fömenu                               *
*********************************************


SET SAFETY OFF
SET TALK OFF
SET STAT OFF
* Hiba esetén HIBA.PRG hibaészlelö és ujraélesztö program inditása

ON ERROR DO HIBA

* A file-okat megosztva használjuk

SET EXCLUSIVE OFF

DO WHILE .T.
   CLEAR
   CLEAR ALL
*  Keret rajzolás

   @  3,4 to 21,76
   @  22,0 to 24,79
   @  4,18 say "          Jegyek eladása"
   @  5,18 say "-------------------------------------"
   @  7,15 say "Választási lehetöségek :"
   @  9,22 say "1. Dátum/elöadás TÖRLÉSe a müsornaptárból"
   @ 11,22 say "2. Ujabb elöadás FELVÉTELe a müsornaptárba"
   @ 13,22 say "3. JegyELADÁs"
   @ 15,22 say "4. Müsornaptár MEGJELENITÉSe"
   @ 17,22 say "0. KILÉPÉS"
   @ 19,15 say "Kérem válasszon :"
   VALASZ=' '
   DO WHILE .NOT. VALASZ$"01234"
     @ 19,43 GET VALASZ
     READ
   ENDDO
   CLEAR GETS
   DO CASE

     CASE VALASZ='0'

*      Kilépés

       SET EXCLUSIVE ON
       CLEAR ALL
       CLEAR
       @ 10,25 SAY "Jegyeladási program vége"
       CANCEL
     CASE  VALASZ='1'

*      Hely/jegy TÖRLÉSe
*      Csak HSZ=ACCESS()=1 hozzáférési szint esetén engedélyezett
```

```
        IF ACCESS()<2
          DO HJ_TOR
        ELSE
          @ 23,2 SAY SPACE(77)
          @ 23,2 SAY "A pénztár számára a TÖRLÉS NEM engedélyezett !!;
                     Üssön le egy billentyüt !!"
          SET CONSOLE OFF
          WAIT
          SET CONSOLE ON
        ENDIF

      CASE VALASZ='2'
*       Hely/jegy FELVÉTELe
*       Csak HSZ=ACCESS()=1 hozzáférési szint esetén engedélyezett

        IF ACCESS()<2
          DO HJ_FEL
        ELSE
          @ 23,2 say SPACE(77)
          @ 23,2 SAY "A pénztár számára a FELVÉTEL NEM engedélyezett !!;
                     Üssön le egy billentyüt !!"
          SET CONS OFF
          WAIT
          SET CONS ON
        ENDIF
       CASE VALASZ='3'

*         Hely/jegy eladás

          DO HJ_ELAD

       CASE VALASZ='4'

*         Müsornaptár MEGJELENITÉSe

          DO HJ_MEGJ

    ENDCASE
ENDDO
CANCEL
```

```
*-----------------------------------------*
*  Program : HIBA.PRG                     *
*                                         *
*  Hibákat kiszürö és ujraélesztö program *
*                                         *
*-----------------------------------------*

DO CASE
  CASE ERROR()=108 .OR. ERROR()=109
     * [108] a file-t már használja valaki
     * [109] a rekordot már másvalaki használja
     RETRY
  CASE ERROR()=1
     * Ismeretlen file
     CLEAR
     @ 5,5 SAY MESSAGE()
     CANCEL
  CASE ERROR()=26 .OR. ERROR()=20
    * [26] az adatbázis nem indexelt
    * [20] a rekord nem szerepelt az indexfile-ban
    SELECT 1
    REINDEX
  CASE ERROR() = 148
    * A server tulterhelt
    RETRY
  OTHERWISE
    CLEAR
    @ 23,2 SAY MESSAGE()
    CANCEL
ENDCASE
RETURN
```

```
*----------------------------------------*
*  Program :HJ_FEL.PRG                   *
*                                        *
*  Elöadások felvétele a müsornaptárba   *
*                                        *
*----------------------------------------*


@ 4,5 CLEAR TO 20,75
@ 15,6 TO 20,74

@ 23,2 SAY SPACE(77)
@ 5,30 SAY "Elöadások felvétele a müsornaptárba"
@ 6,30 SAY "-----------------------------------"

@ 8,9 TO 10,72
@ 15,6 TO 20,74
T_DATUM="88/01/01"
T_HELYEK=1
T_AR=0
@ 16,7 CLEAR TO 19,73
@ 9,12 SAY "Kérem adja meg az elöadás dátumát : " GET T_DATUM PICTURE "88/99/99"
@ 11,7 TO 13,74
@ 12,9 SAY "Kérem adja meg a helyek számát : " GET T_HELYEK PICTURE "99"
@ 12,46 SAY "és egy helyjegy árát : " GET T_AR PICTURE "999"
READ

SELECT 1
USE HELY INDEX DHELY
SEEK T_DATUM
IF .NOT.EOF()
   @ 23,2 SAY SPACE(77)
   @ 23,2 SAY "A dátum már szerepel a müsornaptárban! Üssön le egy billentyüt!"
   SET CONSOLE OFF
   WAIT
   SET CONSOLE ON
ELSE
*  Automatikus file zárolás APPEND  esetén

   APPEND BLANK

*  Rekord zárolás REPLACE számára

   ANS='U'
   DO WHILE UPPER(ANS)='U'
     ZARSZAM=250
     DO WHILE ZARSZAM#0.AND.ANS#'K'
       IF .NOT.RLOCK()
         ZARSZAM=ZARSZAM-1
     * ELSE
         ANS='K'
       ENDIF
     ENDDO
     IF ZARSZAM=0
       ANS=' '
       @ 23,2 SAY ;
       "HELY/JEGYFORGALOM folyamatban!! "U" ujra próbálkozik, egyébként Kilépés"
```

```
      SET CONSOLE OFF
      WAIT TO ANS
      IF UPPER(ANS)#'U'
        ANS='K'
      ENDIF
    ENDIF
  ENDDO

  @ 17,20 SAY T_DATUM+" -én"
  @ 17,35 SAY T_HELYEK PICT "99"

  @ 17,40 SAY "  hely üres."
  REPLACE HELYEK WITH T_HELYEK
  REPLACE DATUM  WITH T_DATUM
  REPLACE AR     WITH T_AR
  UNLOCK
ENDIF
CLOSE DATA
@ 23,2 SAY SPACE(77)
@ 23,10 SAY "Kiván ujabb elöadást felvenni ? (i,n)"
SET CONSOLE OFF
WAIT TO ANS
SET CONSOLE ON
IF UPPER(ANS)='I'
  RETRY
ENDIF
RETURN
```

```
*----------------------------------------*
*  Program :HJ_TOR.PRG                   *
*                                        *
*  Elöadások törlése a müsornaptárból    *
*                                        *
*----------------------------------------*

@ 4,5 CLEAR TO 20,75
@ 15,6 TO 20,74
@ 23,2 SAY SPACE(77)
@ 5,30 SAY "Elöadások törlése a müsornaptárból"
@ 6,30 SAY "-----------------------------------"

@ 8,9 TO 10,72
@ 15,6 TO 20,74
ISMET=.T.
T_DATUM="88/01/01"
T_HELYEK=1
T_AR=0
@ 16,7 CLEAR TO 19,73
@ 9,12 SAY "Kérem adja meg az elöadás dátumát : " GET T_DATUM PICTURE "88/99/99"
READ

* Törlés - PACK kizárólagos megnyitást igényel

SET EXCLUSIVE ON
SELECT 1
USE HELY INDEX DHELY
SEEK T_DATUM
IF EOF()
   @ 23,2 SAY SPACE(77)
   @ 23,2 SAY "A dátum nem szerepel a müsornaptárban! Üssön le egy billentyüt!"
   SET CONSOLE OFF
   WAIT
   SET CONSOLE ON
ELSE
   @ 17,20 SAY T_DATUM+" -én"
   @ 17,35 SAY HELYEK PICT "99"
   @ 17,40 SAY "  hely üres."
   DELETE
   PACK
ENDIF
SET EXCLUSIVE OFF
@ 23,2 SAY SPACE(77)
@ 23,10 SAY "Kiván ujabb elöadást törölni ? (i,n)"
SET CONSOLE OFF
WAIT TO ANS
IF UPPER(ANS)='I'
  RETRY
ENDIF
RETURN
```

```
*-----------------------------------------*
*  Program :HJ_ELAD.PRG                   *
*                                         *
*  Helyjegyek eladása adott dátum szerint *
*                                         *
*-----------------------------------------*


@ 4,5 CLEAR TO 20,75
@ 15,6 TO 20,74
@ 23,2 SAY SPACE(77)
@ 5,30 SAY "        Helyjegyek eladása         "
@ 6,30 SAY "-----------------------------------"

@ 8,9 TO 10,72
@ 15,6 TO 20,74
T_DATUM="88/01/01"
T_HELYEK=1
@ 16,7 CLEAR TO 19,73
@ 9,12 SAY "Kérem adja meg az elöadás dátumát : " GET T_DATUM PICTURE "88/99/99"
READ
SELECT 1
USE HELY INDEX DHELY
SEEK T_DATUM
IF HELYEK=0.OR.EOF()
*  A dátum nem szerepel a müsornaptárban, vagy a helyek száma = 0
   @ 23,2 SAY SPACE(77)
   @ 23,10 SAY "Az adott idöpontra nincs jegy !!  Üssön le egy billentyüt !!"
   SET CONSOLE OFF
   WAIT
   SET CONSOLE ON
ELSE
   ANS='U'

*  Zárolási kisérletek, mig ZARSZAM>0

   DO WHILE UPPER(ANS)='U'
     ZARSZAM=250
     DO WHILE ZARSZAM#0.AND.ANS#'K'
       IF .NOT.RLOCK()
         ZARSZAM=ZARSZAM-1
       ELSE
         ANS='K'
       ENDIF
     ENDDO
     IF ZARSZAM=0
       ANS=' '
       @ 23,2 SAY "HELY/JEGYFORGALOM folyamatban!! "U" ujra próbálkozik,
                  egyébként Kilépés"
       SET CONSOLE OFF
       WAIT TO ANS
       IF UPPER(ANS)#'U'
         ANS='K'
       ENDIF
     ENDIF
   ENDDO
   T_HELYEK=HELYEK
```

```
   @ 17,20 SAY T_DATUM+" -én"
   @ 17,35 SAY T_HELYEK pict "99"
   @ 17,40 SAY " hely üres."
   @ 11,9 TO 13,72
   ANS='I'
   DO WHILE UPPER(ANS)='I'
     @ 12,15 SAY "Hány helyet igényel ? " GET T_HELYEK PICTURE "99"
     READ
     IF T_HELYEK > HELYEK
       @ 23,2 SAY SPACE(77)
       @ 23,10 SAY "Ennyi hely nem áll rendelkezésére! Kiván ismételni? (i,n)"
       SET CONSOLE OFF
       WAIT TO ANS
       SET CONSOLE ON
     ELSE
       REPLACE HELYEK WITH HELYEK-T_HELYEK
       UNLOCK
       ANS='N'
     ENDIF
   ENDDO
ENDIF
@ 23,2 SAY SPACE(77)
@ 23,10 SAY "Kiván ujabb jegyeket eladni ? (i,n)"
SET CONSOLE OFF
WAIT TO ANS
IF UPPER(ANS)='I'
  RETRY
ENDIF
RETURN


*-------------------------------------------------*
*  Program :HJ_MEGJ.PRG                           *
*                                                 *
*  Elöadások dátuma; Szabad helyek száma ; Ar     *
*                                                 *
*-------------------------------------------------*

SET EXCLUSIVE OFF
USE HELY INDEX DHELY
CLEAR
REPORT FORM MUSOR TO PRINT
USE
@ 22,0 TO 24,79
@ 23,20 SAY "Ha befejezte üssön le egy billentyüt !!"
SET CONSOLE OFF
WAIT
SET CONSOLE ON
RETURN
```

13.5. Hálózati parancs és függvénygyűjtemény

13.5.1. Felhasználót segítő információkat szolgáltató parancsok

13.5.1.1. Parancs: DISPLAY STATUS (TO PRINT)

Az aktuális dBase munkafázissal kapcsolatban szolgáltat értékes információkat. A megjelenített információegyüttes az alábbi képet mutatja:

```
Select area:  1, Database in Use: F:hely.dbf    Alias: HELY
    Master index file:  F:dhely.ndx  Key: datum
         Lock:

Currently Selected Database:
Select area:  2, Database in Use: F:nevek.dbf    Alias: NEVEK
    Master index file:  F:nevek.ndx  Key: szol
         Lock:

File search path:
Default disk drive: F:
Print destination:  LPT1:
Margin =     0
Current work area =    2

ALTERNATE  - OFF   DELETED    - OFF   EXCLUSIVE  - ON    PRINT      - OFF
BELL       - ON    DELIMITERS - OFF   FIELDS     - OFF   SAFETY     - ON
CARRY      - OFF   DEVICE     - SCRN  FIXED      - OFF   SCOREBOARD - ON
CATALOG    - OFF   DOHISTORY  - OFF   HEADING    - ON    STATUS     - OFF
CENTURY    - OFF   ECHO       - OFF   HELP       - ON    STEP       - OFF
CONFIRM    - OFF   ENCRYPTION - ON    HISTORY    - ON    TALK       - ON
CONSOLE    - ON    ESCAPE     - ON    INTENSITY  - ON    TITLE      - ON
DEBUG      - OFF   EXACT      - OFF   MENU       - ON    UNIQUE     - OFF

Programmable function keys:
F2  - assist;
F3  - list;
F4  - dir;
F5  - display structure;
F6  - display status;
F7  - display memory;
F8  - display;
F9  - append;
F10 - edit;
```

13.5.1. ábra

Az információ összetétele:

megnyitott adatbázis file-ok:

- adatbázis file neve
- munkaterület
- alias
- adatbázis file-ok kapcsolatai

Hálózati környezetben:

- adatbázis file-ok zárolási állapota
- adatbázis file-ok rekordjainak zárolási állapota

megnyitott index file-ok:

- index file-hoz tartozó indexelési kulcs
- Aktuális file elérési út (path)
- Aktuális meghajtó
- Kimeneti egység kijelölés (SET DEVICE TO SCREEN: képernyőre ; TO PRINT: nyomtatóra)
- Bal margó beállítás
- SET ON/OFF típusú parancsokkal vezérelhető kapcsolók allapota. Hálózati környezetben az EXCLUSIVE és az ENCRYPTYON kapcsolók állapota is megjelenik.

DISPLAY STATUS TO PRINT esetén az információ az aktuális nyomtatóra kerül. Az aktuális nyomtató SET PRINTER TO paranccsal kijelölt hálózati vagy egy helyi nyomtató. Az információ-megjelenítés időszakonként megszakad, s csak egy billentyű leütése után folytatódik.

13.5.1.2. Parancs: LIST STATUS (TO PRINT)

A parancs szerepe és az általa szolgáltatott információk azonosak a DISPLAY STATUS parancs megjeleníttette adatokkal.
Az egyetlen különbség, hogy a DISPLAY STATUS megjelenítés időszakonként megszakad, míg a LIST STATUS megjelenítés folyamatos.

13.5.1.3. Parancs: DISPLAY USERS

A hálózatba kapcsolt, dBASE ADMINISTRATOR -t használó állomások bejelentkezési neveit jeleníti meg a következő formában:

```
type alma.txt

  Computer name
  -------------
  TIBOR
  GABOR
> HUNOR

F>
```

13.5.2. ábra

A " " jelöli az aktuális állomás nevét (amelyiken a parancsot kiadtuk).

13.5.2. Adatbázis védelem

13.5.2.1. Parancs: SET ENCRYPTION ON/OFF

Ez a parancs az ujonnan létrehozott file-ok titkosításáról rendelkezik. Alapértelemzés: SET ENCRYPTION ON. Egy titkosított file olyan formában kódolt, hogy tartalma az illetéktelen felhasználó előtt rejtve marad. Érvényes, elfogadott bejelentkezés esetén a dekódolás a felhasználói állomásokon automatikusan megtörténik.

A bejelentkezés három eleme:

- a csoport neve
- a felhasználó neve
- a jelszó

Dekódolás után a file tartalma kimásolható kódolatlan formában, a SET ENCRYPTION OFF parancs kiadása után.

A titkosítás kikapcsolt állapota esetén a COPY, JOIN, SORT és TOTAL parancsok titkosítatlan file-okat eredményeznek.

A COPY TO parancs opciók, DELIMITED, DIF, PSF, SDF, SYLK, WKS csak SET ENCRYPTION OFF esetén érvényesek.

A CREATE révén létrehozott file titkosítását a PROTECT-tel valósíthatjuk meg. A másolt file tartalma a felhasználói hozzáférési szint és a file/mező privilégiumok viszonyának függvénye.

Az egyfelhasználós környezet titkosítatlan file-okat használ. Így többfelhasználós környezet titkosított file-jait – az előbb elmondottak alapján – a következő módszerrel menthetjük át egyfelhasználós környezetbe:

```
USE file1
SET ENCRYPTION OFF
COPY TO file2
```

A file2 már nem titkosított.

Megjegyzés: A MODIFY STRUCTURE parancs titkosított file-ok esetén nem alkalmazható.

13.5.2.2. Parancs: LOGOUT

Ez a parancs az aktuális felhasználó kijelentkezésének és egy másik felhasználó bejelentkezésének feladatát látja el.

A LOGOUT hatására a képernyő törlődik és egy új bejelentkezési képernyő jelenik meg, amely csoport- és felhasználónév, valamint jelszó bevitelét várja. A PROTECT által felépített védelmi rendszer ellenőrzi a bejelentkezés érvényességét, majd a felhasználóhoz rendeli a PROTECT révén előzőleg meghatározott hozzáférési szintet. A LOGOUT minden nyitott file-t lezár.

Ha a PROTECT használatára még nem került sor és a DBSYSTEM.DB file nem létezik, a LOGOUT parancs értelmetlen és figyelmen kívül marad.

13.5.3. Paraméter beállítás

13.5.3.1. Parancs: SET EXCLUSIVE ON/OFF

A parancs az adatbázis file megnyitásának módját határozza meg. Alapértelemezés szerint minden file megnyitás kizárólagos, vagyis SET EXCLUSIVE ON.

Ha egy felhasználó egy file-t kizárólagosan nyit meg, az adott file, lezárásáig, minden más felhasználó számára hozzáférhetetlen marad. Ilyen esetben a file/rekord zárolás fölösleges.

SET EXCLUSIVE OFF parancs kiadása után a USE-al megnyitott file-ok megosztva vehetők igénybe, a file több felhasználó számára hozzáférhető. A file/rekord zárolást célszerű használni.

Példa:
```
SET EXCLUSIVE OFF
SELECT 1
USE KOZOS INDEX KOZOS
SET EXCLUSIVE ON
SELECT 2
USE CSAKEN INDEX CSAKEN
```

A KOZOS file több felhasználó számára hozzáférhető, mig a CSAKEN kizárólagos megnyitása minden más felhasználó file hozzáférését lehetetlenné teszi.

13.5.3.2. Parancs: SET PRINTER TO

Ez a parancs a dBASE nyomtató kimenetének helyi (lokális) vagy hálózati nyomtatóra való átirányítását végzi.

Három formája ismeretes:

a. Hálózati nyomtatóra irányítás IBM hálózati környezetben

SET PRINTER TO \\<számítógép_neve>\<nyomtató_neve> = <célnyomtató>

paranccsal történik.

A <számítógép neve> a felhasználói állomás hálózatban szereplő neve, míg a <nyomtató neve> a hálózati a nyomtató neve. Az előbbiekben vázolt két elnevezés IBM hálózati környezetben rendelhető számítógéphez és nyomtatóhoz. A <cél nyomtató> azt a DOS nyomtatót azonosítja, amelyiket hálózati nyomtatóval helyettesítünk. Értéke tehát LPT1, LPT2 vagy LPT3 .

A Novell hálózati környezetben a hálózati nyomtatóra irányítás az alábbi paranccsal történhet:

SET PRINTER TO \\SPOOLER

A SPOOLER a hálózati nyomtatót jelöli.

b. Helyi nyomtatóra irányítás

SET PRINTER TO <célnyomtató>

Helyi egységet jelöl ki nyomtatóként. A (célnyomtató) lehet a párhuzamos kimenetre csatlakoztatott nyomtató LPT1, LPT2, LPT3, vagy soros kimenetre csatolt COM egység COM1, COM2.

A SET PRINTER TO LPTn/COMn a helyi nyomtatóra SPOOL típusú kimenetet biztosít.

c. SET PRINTER TO

A nyomtató SPOOL file kiürítését és az alapértelmezés szerinti nyomtató kijelölését végzi.

A SPOOL file-ba gyűjtött adatok a SET PRINT hatására kerülnek nyomtatásra.

13.5.4. Adatbázis kezelés

13.5.4.1. Parancs:

USE file_nev INDEX indexfile_nevek EXCLUSIVE ; ALIAS alias_nev

Feladata a file_nev adatbázis file és indexfile-jainak kizárólagos megnyitása. Az EXCLUSIVE kulcsszó állhat a file_nev előtt, után vagy a parancs végén.

13.5.5. Zárolással kapcsolatos parancsok és függvények

13.5.5.1. Függvény: FLOCK()

A FLOCK feladata az adott munkaterületen megnyitott file zárolása. Sikeres zárolás esetén, az FLOCK logikai igaz (T.) ellenkező esetben logikai hamis (F.) értéket ad vissza. A file zárolás akkor járhat sikerrel ha az adott file-t előzőleg más felhasználó nem zárolta. Zárolt file zárolási kisérletei sikertelenek maradnak, a zárolás feloldásáig.

Az FLOCK() segítségével elejét vehetjük a file hozzáférések ütközésének. A file hozzáférések akkor ütköznek ha ugyanahhoz a file-hoz egyidejűleg több felhasználó próbál hozzáférni.

A file zárolás feloldása az UNLOCK paranccsal történik, vagy a file lezárási művelet eredménye. Az FLOCK alkalmazása akkor javasolt, ha az adott eljárás sok rekord és index file változtatását célozza. Gondoljunk csak a REPLACE ALL vagy batch típusú parancsokra.

13.5.5.2. Függvény: RLOCK()/LOCK()

A függvény feledata a rekord zárolás. A zárolás sikerét logikai igaz (.T.) visszaadott érték jelzi. Zárolt rekord zárolása lehetetlen, a függvény visszaadott értéke ilyenkor logikai hamis (.F.) lesz.

A rekord zárolás, minden olyan esetben ajánlott, mikor több felhasználó veszi igénybe ugyanannak a file-nak rekordjait. Egy rekord zárolása mellett minden más rekord hozzáférhető marad.

A következő parancsok hálózatban megkövetelik a rekord zárolást:

@ sor,oszlop GET "mezo_nev"
READ
REPLACE
DELETE

Egy adatbázis tranzakció, művelet általában három részből áll:

- adatolvasás
- adatváltoztatás
- adat(vissza)írás

A rekord zárolás megakadályozza több tranzakció ütközését, és így az adatbázis integritásának megsértését.

13.5.5.3. Parancs: UNLOCK (ALL)

A file és rekord zárolás feloldásának feladatát látja el.

Az UNLOCK az aktuális rekord vagy file zárolását, az UNLOCK ALL minden zárolt file és rekord zárolását feloldja.

RLOCK() vagy LOCK() után, az UNLOCK rekord zárolását FLOCK() után file zárolását oldja fel.

13.5.6. Más, programozást segítő parancsok és függvények.

13.5.6.1. Parancs: RETRY

A RETRY parancs hatására az adott program vagy eljárás befejeződik, a vezérlés pedig visszakerül a hívóprogram adott eljárást indító parancssorába, és nem az ezt követő parancshoz, mint az a RETURN esetében történik. A RETRY parancs általában az ON ERROR indította eljárásokban használatos. A dBASE program elején ON ERROR "eljaras_nev" parancs hatására minden későbbi hiba az "eljaras_nev" hibakezelő program indítását eredményezi. A hibakezelést követő újraélesztés a RETRY elsőrendű feladata. A hibát kiváltó parancssor a művelet sikeres végrehajtásáig, vagy időzítő eljárás beépítésével adott ideig ismételhető.

A RETRY parancs a későbbiekben ismertetett ERROR() És MESSAGE() függvények visszaadott értékeit nullázza.

13.5.6.2. Függvény: ACCESS()

A felhasználó hozzáférési szintjével tér vissza.Ennek értéke 1-8 közötti egész szám. ACCESS() segítségével szelektív adatvédelmet valósíthatunk meg. A PROTECT -tel megadott hozzáférési szint ellenőrzésével, adott felhasználók számára programágak, eljárások végrehajtását tehetjük lehetetlenné. Egyfelhasználós környezetben az ACCESS mindig 0-val tér vissza.

13.5.6.3 Függvény: ERROR()

Az ERROR() az utoljára észlelt hiba kódjával tér vissza. Az ERROR() függvény hibakezelő és újraélesztő eljárások alapkfüggvénye, hiszen a hiba azonosítása után a megfelelő válaszlépések megtételére is lehetőség nyílik. Az ERROR() csak az ON ERROR indította hibakezelő eljárásokban használható. A hibakezelő eljárás RETURN és RETRY parancsai nullázzák a hibakódot, és így az ERROR visszaadta értéket is törlik.

13.5.6.4 Függvény: MESSAGE()

A MESSAGE() a hibakódot azonosító üzenetet adja vissza. Lehetőség nyílik a hibaüzenet kiíratására vagy változóba mentésére.

Ez a függvény akárcsak az ERROR() csak az ON ERROR indította eljárásokban használható. A RETRY és RETURN parancsok nullázzák a hibakódot és törlik a hozzá tartozó karaktersorozatot.

14. CLIPPER

14.1 A dBase III PLUS és a Clipper

14.1.1 A Clipper, mint adatbáziskezelő programcsomag

A Clipper egy önálló rendszer, amely mindazon eszközöket magában foglalja, amelyek egy felhasználói program fejlesztéséhez szükségesek. A Clipper-rel minden egyéb software felhasználása nélkül – kivéve a szövegszerkesztőt – létrehozhatók, és lefordíthatók a programok. A hibamentes Clipper programok lefordítása és szerkesztése után a vérehajtható (.EXE kiterjesztésű) file-ok bármely számítógépen futtathatók, amely a PC DOS 2.0, vagy annál fejlettebb változatát (illetve hálózati alkalmazás esetén a PC DOS 3.1, vagy annál fejlettebb változatát) tartalmazza.

14.1.2 A Clipper, mint dBASE III PLUS-fordító

A Clipper logikai felépítése – ahol csak lehetséges – követni igyekszik a dBASE III PLUS programozási nyelvet. A dBASE III PLUS nyelven megírt alkalmazói programok tehát némely fenntartásokkal sikeresen lefordíthatók a Clipper-rel.

A fordításhoz a program módosítására van szükség, ha:

- a program tartalmaz olyan dBASE III PLUS parancsot, amelyet a Clipper eltérő módon, vagy egyáltalán nem támogat.
- a programban olyan makro van, amely parancsot tartalmaz.

(A különbségek részletezését a 14.4. fejezet tartalmazza.)

14.1.3 Az interpreter és a fordító közötti különbségek

Az interpretált nyelveken megírt program különböző parancsok, és függvények soraiból áll. Amikor a program fut, az interpreter sorról-sorra lépked, és ha nem észlel hibát, minden egyes sort külön-külön lefordít gépi kódra. Ha az interpreter az átalakítást megelőzően hibát talál, akkor a végrehajtás megszakad, és a képernyőn hibaüzenet jelenik meg. A programot a hiba javítása után lehet csak újra indítani. Ez az eljárás tart mindaddig, amíg a program hibamentes nem lesz. Sajnos a tökéletes programok is ugyanennek az időrabló, minden egyes sort újra elemző eljárásnak lesznek alávetve minden futás alkalmával. Nincsen lehetőség a befejezett programok gépi nyelvű kódjának tárolására, vagy a folytonos ellenőrzés kikapcsolására, pedig ez jelentősen növeli a program végrehajtásához szükséges időt.

A fordítók használatával – az interpreterekhez hasonlóan – az ellenőrzés sorról-sorra történik, azonban a hibás sor számát és a hozzá tartozó hibaüzenetet kiírva a képernyőre általában nem szakad meg a fordítás, hanem a program végének eléréséig folyamatos. Eközben a fordító a forráskódot lefordítja tárgykóddá (gépi nyelvű), és a sorokat egy .OBJ file-ban tárolja. Ez a tárgykód a szerkesztés előtt nem futtatható. (Természetesen a hibás programok tárgykódjának megőrzése értelmetlen, a szöveg-

szerkesztővel javítsuk a programot, és fordítsuk újra.) A szerkesztés alkalmával kapcsoljuk hozzá a tárgykódhoz az összes futáshoz szükséges rutint, és önállóan végrehajtható .EXE file-t kapunk. Az .EXE file-ok futtatását közvetlenül az operációs rendszerből, a programnév begépelésével lehet elérni.

A fordítók további előnye, hogy előzőleg lefordított eljárások is hozzászerkeszthetők a programokhoz. Külön figyelemre méltó, hogy ezeket a rutinokat írhatjuk más nyelven is (a hívási konvenciók betartásával) – például C, vagy Assembly nyelven –, és külön fordíthatjuk.

Megjegyzendő, hogy a végrehajtható (.EXE) file jelentősen nagyobb, mint a tárgymodul (.OBJ) éppen azért, mert tartalmazza a futáshoz szükséges eljárásokat, és a főprogramból hívott, előzőleg lefordított programokat, vagy felhasználói eljárásokat is.

A szerkesztés folyamán előfordulhatnak hibaüzenetek. (A szintaktikus hibákat a fordító felderíti!) Ilyen lehet pl.: megkettőzött, vagy meghatározatlan hivatkozás. Ezeket a hibákat vagy a forráskód javításával, vagy a szerkesztés során kihagyott modulok hozzáadásával lehet kiküszöbölni. Ezután – szükség szerint – fordítsuk, és/vagy szerkesszük újra a programot.

A program végrehajtása közben 'run-time' (futás közbeni) hibákat észlelhetünk, amelyek a program futásának megszakadását okozhatják. Egy egyszerű run-time hiba lehet például ha egy nemlétező adatbázis-file használatát kíséreljük meg. Ilyenkor a képernyőn hibaüzenet jelenik meg. A forrásprogram módosítása rendszerint nem szükséges, a hibát a file létrehozásával, vagy az azt tartalmazó floppy behelyezésével megszüntethetjük. A program így újra futtatható lesz. (A fordítás, szerkesztés és futtatás közben előforduló hibaüzeneteket a 14.6 fejezet tartalmazza.) Az overlay-technika alkalmazásával elérhetjük, hogy ha a program túlságosan nagy a rendelkezésre álló RAM-területhez képest, akkor a .EXE file-t szegmensekre osztva hívjuk be. A szegmensek szükség szerint töltődnek be ugyanarra a közösen használt memóriaterületre. (Az overlay-k használatát a 14.2.2.5 fejezet írja le.)

A Clipper-rel fordított program futtatásához tehát nincsenek további követelmények. Nem szükséges hozzá interpreter, a forráskód, az object file, sőt még egy Clipper-ről készült másolat sem. Elegendő a végrehajtható (.EXE) file és a futás közben használt adatbázis- (.DBF), index- (.NTX), jelentés szerkezeti- (.FRM), cimke- (.LBL), vagy memória- (.MEM) file-ok jelenléte.

14.1.4 A Clipper előnyei

- Az interpretív nyelveknél számottevően gyorsabb végrehajtás
- Valamennyi programhiba azonosítása egyidőben
- A forráskód ellenőrzése minden fordításnál
- A forráskód abszolut védelme
- A Clipper-rel fordított programok önálló terjeszthetősége (egyéb softwareszükséglet nélkül)
- Az eljárások és függvények elhelyezhetősége a hívó programmal azonos file-ban

- Max. 2048 aktív memóriaváltozó használata
- Adatbázisonként max. 1024 adatmező használata
- Korlátlan számú C vagy Assembly nyelven írt külső program hívhatósága
- Rekurzív, vagy beágyazott makrok használata
- Ciklusok létesítése a BASIC-éhez hasonló FOR ... NEXT utasítással
- Tömbök kezelése
- A CLIPPER nevű public változó használata a Clipper programok interpreter alatti futtatásához
- Utility-k az adatbázis-, index-, cimke- és jelentés szerkezeti file-ok létrehozásához
- Szövegérzékeny help-szolgáltatások beépítésének lehetősége
- Teljeskörű nyomkövető-szolgáltatások

14.1.5 A gépi környezettel szemben támasztott követelmények

14.1.5.1 Hardware-feltételek

A programok Clipper-rel történő fordítása olyan számítógépet kíván, amely rendelkezik az alábbi tulajdonságokkal:

- IBM PC, XT vagy AT, illetve ezekkel 100%-ig kompatibilis számítógép
- A PC DOS 2.0 illetve annál fejlettebb változata
- Legalább 256 kByte RAM
- 1 db hajlékony- és 1 db fixlemez egység, vagy 2 db hajlékonylemez egység
- Hasznos lehet még egy nyomtató, 8087 vagy 80287-es tipusú matematikai koprocesszor

A Clipper-rel lefordított programok futtatásához olyan számítógéppel kell rendelkeznünk, amely a következő feltételeknek megfelel:

- IBM PC, XT vagy AT, illetve ezekkel 100%-ig kompatibilis számítógép
- A PC DOS 2.0 illetve annál fejlettebb változata
- Legalább 256 kByte RAM
- Egy olyan szamítógép, amely az ANSI szabványnak megfelelően működik

14.1.5.2 A DOS-környezet módosításai

A DOS-szal kapcsolatos megfontolások csak a lefordított és szerkesztett alkalmazói programokat érintik. A DOS-környezet módosítása nem befolyásolja a Clipper fordító és a PLINK86 program szerkesztő működését.

14.1.5.2.1 A DOS parancs-processzor

Ha olyan programot futtatunk, amely – a "RUN" parancs segítségével – külső programot is hív, a DOS-nak el kell érnie a DOS parancs-processzort (COMMAND.COM). Ha a program és a DOS parancs-processzor nem ugyanabban a könyvtárban található, akkor a DOS "SET" parancsával meg kell határoznunk, hogy hol keresse a DOS a parancs-processzort:

Írjuk be a "SET" parancsot, és üssük be az Enter-t. A képernyőn megjelenik a DOS-környezet pillanatnyi állapota.

A "COMSPEC=" utasítás lesz az, amely meghatározza, hogy a DOS hol keresse a COMMAND.COM-t. Ennek megváltoztatásához írjuk be a következő DOS parancsot:

C>SET COMSPEC=<d:\út\COMMAND.COM>

ahol: "d:" a meghajtó azonosítója
"út" a COMMAND.COM elérési útja

(A "PATH" parancs használatáról a DOS kézikönyv tájékoztat.)

14.1.5.2.2 A DOS FILES és BUFFERS paraméterei

Az adatbázisok kezelése intenzívebb lemezhasználatot igényel, mint más tipusú programok. A DOS megengedi a környezet olyan irányú módosítását, amely elősegíti az intenzív lemezhasználatot.

A DOS "FILES" paramétere azt határozza meg, hogy hány file lehet nyitva egyidőben. (A PC-DOS alapértelmezése 8.) A Clipper-rel fordított program maximum 15 file-t tud egyszerre megnyitni.

A DOS "BUFFERS" paramétere azt mutatja meg, hogy a DOS hány puffert használ egy lemezes írás-olvasási folyamat során. (A PC-DOS alapértelmezése 2.) Az ellenőrzések azt mutatták, hogy a Clipper számára a megfelelő érték 8.

Ezek a paraméterek a CONFIG.SYS file létrehozásával, illetve megváltoztatásával állíthatók. Minden alkalommal, amikor a számítógépet bekapcsoljuk, vagy újra töltjük, a DOS megkeresi a CONFIG.SYS file-t, és a környezetet ennek megfelelően módosítja. A CONFIG.SYS file-nak tartalmaznia kell a következő utasításokat:

```
FILES=20
BUFFERS=8
```

(Azaz 20 file lehet nyitva egyidőben, ebből 5-t a DOS használ.)

14.1.5.2.3 Az ANSI terminál támogatás

A Clipper tartalmaz egy ANSI terminál támogató programot (ANSI.OBJ) olyan számítógépek, vagy felhasználói programok számára, amelyek ezt igénylik. Az ANSI.OBJ file a szerkesztéskor a program szerkesztővel hozzáfűzhető a felhasználói programhoz. A működéshez azonban szükséges az ANSI periféria kezelő (ANSI.SYS) installálása is. Ezt úgy érhetjük el, hogy a CONFIG.SYS file-ba beírjuk a következő utasítást:

```
DEVICE=ANSI.SYS
```

Ezután a rendszert bekapcsolva, vagy újratöltve az ANSI periféria kezelőt be is töltöttük a rendszerbe.

14.2 Fordítás és szerkesztés

14.2.1 Fordítás

A Clipper fordító program a dBASE III PLUS programnyelven írt forrásprogramokat (amelyeket a .PRG kiterjesztés azonosít) átalakítja tárgykóddá (.OBJ kiterjesztéssel). Ez a modul a program szerkesztővel (PLINK86) szerkeszthető végrehajtható file-lá (amely a .EXE kiterjesztést kapja).

A Clipper-rel programok, eljárások, a felhasználó által definiált függvények és a formátum file-ok fordíthatók külön-külön, vagy egy csoportban. Ez attól függ, hogy hogyan stuktúráltuk a programot, és hogyan kívánjuk struktúrálni a végrehajtható file-t.

A programok, eljárások, és felhasználói függvények egyaránt létrehozhatók sorszerkesztő, vagy szövegszerkesztő segítségével. Irhatunk adatbázis-, jelentés szerkezeti- és cím-file-okat a Clipper-hez tartozó utility programok felhasználásával is.

A Clipper fordító a következő paranccsal futtatható:

C>CLIPPER [d:][\út\]<file-név>
ahol: "d:" a meghajtó azonosítója
"út" az elérési út
"file-név" a programfile neve a .PRG kiterjesztés nélkül

A fenti parancs eredményeképpen az aktuális könyvtárban egy tárgykódú (object) file keletkezik az eredeti programnévvel azonos néven, de .OBJ kiterjesztéssel.

A programban hívott összes eljárást és formátumot a Clipper egyidejüleg lefordítja, és azok a tárgymodul részeivé válnak. Ezért a Clipper-nek el kell érnie az összes szükséges file-t. Ha nem alkalmazunk .CLP file-t (ld.:14.2.1.2), a Clipper a hívott file-okat a következő helyeken keresi:

- először ellenőrzi, hogy nem része-e a főprogramnak;
- majd kikeresi a "SET PROCEDURE TO <otherfile>" parancsot, és az otherfile .PRG nevű file-t nézi végig;
- végül a fix lemez elérhető részein keresi. (Az "APPEND" DOS paranccsal állítható be.)

Ha akkor sem találja, hibaüzenetet küld. (A fordítás hibaüzeneteit a 14.6.1. fejezet részletezi.)

Szükség esetén használható a DOS output standard átirányító funkciója a fordító képernyőre írt üzeneteinek átirányítására egy lemezes file-ba, amely később megtekinthető, vagy kinyomtatható:

C>CLIPPER <file-név1> [> <file-név2>]
ahol: "file-név1" a fordítandó program neve
"file-név2" az output file neve
">" aktiválja a DOS átirányító funkcióját

14.2.1.1 A Clipper fordító speciális futási paraméterei

A Clipper fordító megengedi három opcionális kapcsoló állítását futás közben. Ezeket külön-külön, vagy együttesen elhelyezhetjük a parancsba az alábbiak szerint:

C>CLIPPER <file-név> [[-n]/[-l]/[-s]]

ahol: "file-név" a fordítandó program neve
"-n","-l","-s" a kapcsolók (kisbetűvel kell írni, másképp a fordító figyelmen kívül hagyja!)

14.2.1.1.1 Bővített eredeti kód (-n)

Ez az üzemmód egy optimalizált kódot eredményez, amely kissé növeli ugyan a tárgymodul méretét, de a program futásának sebessége is megnő.

14.2.1.1.2 Sorszámok nélkül (-l)

Ez az üzemmód kihagyja a tárgymodulból a forráskód sorszámait. A tárgykódú modul mérete csökken, hiszen minden parancsot tartalmazó sor hossza 3 Byte-tal kisebb lesz. FIGYELMEZTETÉS: A Clipper nyomkövető működéséhez szükséges a sorszámok ismerete, így ezen opció a nyomkövető használata mellett kerülendő!

14.2.1.1.3 A program szintaktikus ellenőrzése (-s)

Ez az üzemmód letiltja a tárgymodul létrehozását, ezért akkor használatos, ha a program szintaktikus hibáit akarjuk csak kiszűrni.

14.2.1.2 A név.CLP file-ok használata

A Clipper lehetőséget kínál a program-, eljárás-, formátum- és felhasználói függvény-file-ok elkülönített csoportokban történő fordítására.

Többnyire az egy felhasználói proramhoz tartozó eljárás-, formátum- és függvény-file-okat a felhasználói programéval azonos könyvtárban helyezzük el a fix lemezen, vagy ugyanarra a hajlékony lemezre mentjük. A főprogram fordítása alatt a Clipper meg fogja kísérelni az összes .PRG file lefordítását, amelyre a programok hivatkoznak. Több okból is hasznos lehet azonban a file-ok egyes csoportjait önálló tárgymodulokba fordítani. Például célszerű lehet egy egyedülálló file, vagy a file-ok egy csoportjának külön fordítása hibakeresési céllal, illetve overlay-k létrehozása érdekében. Másfelől szükség lehet a .CLP file-ok alkalmazására a következő okokból is: amikor a Clipper fordít egy programot, két táblázatot létesít, egyiket a program által használt konstansok, másikat a szimbólumok számára. Ezeknek a konstansoknak, illetve szimbólumoknak a száma a táblázaton belül nem lehet korlátlan. Ha a konstans-tábla mérete meghaladja a 64 kByte-t, hibaüzenetet kapunk. A probléma megoldása az lehet, hogy a .CLP file segítségével néhány program- és/vagy eljárásfile-t külön fordítunk. Az elkülönített tárgymodulokat a szerkesztővel fűzhetjük össze egy végrehajtható file-lá.

A .CLP file-okat létrehozhatjuk tetszőleges, szövegszerkesztővel, vagy a DOS "COPY CON" parancsával. A lefordítani kívánt formátum file-okat át kell nevezni a .CLP file megírása előtt úgy, hogy a .FMT kiterjesztéseket .PRG -re változtatjuk. A .CLP file a lefordítandó file-k felsorolását fogja tartalmazni a .PRG kiterjesztés nélkül. (Fontos megjegyezni, hogy a .CLP file-ba nem írhatjuk bele a meghajtó azonosítóját, illetve az elérési utat, azaz a file-oknak a lemezegységnek ugyanabban az alkönyvtárában kell elhelyezkedniök.)

Példa a .CLP file létrehozására a DOS "COPY CON" parancsával:

```
C>COPY CON: Program.clp
Pro1
Pro2
Pro3
```

végül F6 és Enter

A fenti módon egy PROGRAM.CLP nevű file-t hozunk létre, amely három program-file-t tartalmaz, a PRO1.PRG-t, a PRO2.PRG-t és a PRO3.PRG-t

(A "COPY CON" parancs hsználatáról a DOS kézikönyv tájékoztat.)

A programok fordítása ezután a következő paranccsal történhet:

```
C>CLIPPER @<file-név>
```

ahol: "@" azt jelzi, hogy az ezután következő file-név olyan file-t azonosít, amely a fordítandó programok neveinek felsorolását tartalmazza
"file-név" az általunk létrehozott .CLP file neve (a kiterjesztés nélkül)

A fordítás eredménye ezek után egy olyan tángymodul lesz, amelynek neve megegyezik a .CLP file nevével, a kiterjesztése pedig .OBJ. Például a

```
C>CLIPPER @Program
```

utasítás eredménye a PROGRAM.OBJ modul lesz.

14.2.2 Szerkesztés

A program szerkesztő feladata, hogy egy vagy több tárgykódú modulból a DOS alatt futtatható file-t hozzon létre. Az object modulokat a .OBJ kiterjesztés azonosítja, a futáshoz szükséges eljárásokat pedig a .LIB kiterjesztés. A futtatható file kiterjesztése .EXE lesz.

A végrehajtható file-okat szerkeszthetjük bármely szerkesztővel. A DOS szerkesztője – bár némileg gyorsabb a PLINK86-nál – overlay-k előállítására nem alkalmas. (A DOS Linker használatáról a DOS kézikönyv ad leírást.)

A PLINK86 program szerkesztő kifejezetten a Clipper programok szerkesztésére alkalmas, de természetesen szerkeszthetők vele más fordítóval készült modulok is – feltéve, hogy egy Clipper program hívja azokat. Ennél fogva használata kényelmesebb, hiszen automatikusan megkeresi a Clipper könyvtárat (CLIPPER.LIB) és az

overlay könyvtárat (OVERLAY.LIB), valamint használatával alkalmazható az overlay-technika is.

Három módszer létezik a PLINK86 alkalmazásakor:

1.) Egyetlen parancssor használata
2.) Interaktív módszer
3.) A .LNK file használata

Figyeljünk arra, hogy a szerkeszteni kívánt file-ok közül az első mindíg Clipper-rel fordított legyen, másképp a PLINK86 hibát jelez. (A szerkesztés közben előforduló hibaüzeneteket a 14.6.2. fejezet tartalmazza.)

14.2.2.1 A PLINK86 parancsai

A PLINK86-nak a következő parancsok segítségével adhatjuk meg a futási paramétereket:

14.2.2.1.1

A szövegeket, vagy megjegyzéseket tartalmazó sorok megjelölésére szolgál (mindíg a sor első karakterhelyén kell állnia). A PLINK86 a sort kihagyja.

14.2.2.1.2 BEGINAREA

Rövidíthető "BEGIN"-ként. Az overlay-k létrehozásakor használjuk, minden overlay-területnek a "BEGINAREA" paranccsal kell kezdődnie, és az "ENDAREA" paranccsal kell végződnie.

14.2.2.1.3 DEBUG

A programhibák kiküszöböléséhez ad segítséget például azzal, hogy kiírja a képernyőre a programfutás során éppen betöltődő overlay nevét.

14.2.2.1.4 ENDAREA

Rövidíthető "END"-ként. Az overlay-k létrehozásakor használjuk, minden overlay-területnek a "BEGINAREA" paranccsal kell kezdődnie, és az "ENDAREA" paranccsal kell végződnie.

14.2.2.1.5 FILE

Rövidíthető "FI"-ként. Az object modulokat azonosítja a PLINK86 számára, a file-név után az .OBJ kiterjesztést nem kell megadni. Egy "FILE" parancs után több file-név is megadható vesszővel elválasztva.

14.2.2.1.6 LIBRARY

Rövidíthető "LIB"-ként. A futáshoz szükséges eljárások és függvények könyvtárait azonosítja a PLINK86 számára, a feltételezett kiterjesztés .LIB – ha nincs más meg-

adva. A PLINK86 automatikusan megkeresi a CLIPPER.LIB és overlay-k használata esetén az OVERLAY.LIB file-okat, így – ha csak ezt a két könyvtárat használjuk – a parancs használata nem szükséges.

14.2.2.1.7 MAP

A parancs egy .MAP kiterjesztésű file-ba rögzíti a memória használatát, nagyságát és elhelyezkedését. A "MAP" parancsot az alábbiak szerint használjuk:

MAP = <file-név> [[A]/[,G]/[,M]/[,S]]

ahol: "file-név" a keletkező .MAP file neve
"A","G","M","S" a lehetséges kapcsolók
A: (all) felsorolja a programrészeket a feldolgozás sorrendjében, és a programrészek szegmenseit a memóriacímek sorrendjében
G: (global) felsorolja a public változókat és memóriacímüket
M: (modules) ez adja a legrészletesebb információkat; felsorol minden program modult a feldolgozás sorrendjében a memóriacímükkel és nagyságukkal, a modulokban előforduló szegmenseket, és a szegmenseket tartalmazó változókat a címükkel és nagyságukkal.
S: (sections) felsorolja az összes program-szekciót abc-sorrendben a rájuk vonatkozó információkkal együtt. A kapcsolók közül egy, vagy több használható egyidőben, ha nem adunk meg egyet sem, az alapértelmezés "A" lesz.

14.2.2.1.8 OVERLAY

Meghatározza azon szegmenscsoportok nevét, amelyek az overlay fázisban maradnak. (Az overlay-technika leírását ld. a 14.2.2.5. fejezetben.)

14.2.2.1.9 OUTPUT

Az utána következő file-név lesz a végrehajtható modul neve. Ha ez a parancs hiányzik, a végrehajtható modul neve azonos lesz a "FILE" után elsőként szereplő tárgymodul nevével.

14.2.2.1.10 SECTION

Ez a parancs az utána következő .OBJ file-t úgy azonosítja, hogy azt egy belső overlay fogja tartalmazni.

14.2.2.1.11 SECTION INTO

Ez a parancs az utána következő .OBJ file-t úgy azonosítja, hogy azt egy külső overlay fogja tartalmazni.

14.2.2.1.12 VERBOSE

A képernyő utolsó sorában megjeleníti a futás során éppen folyó műveleteket. Használata jelentősen növeli a végrehajtási időt.

Ne használjuk akkor, ha a DOS output átirányító funkcióját használjuk, s a szerkesztő outputját valamely lemezes file-ba irányítottuk.

14.2.2.2 A PLINK86 hívása egyetlen parancssorban

A szerkesztés legegyszerűbb módja, ha a szükséges információkat a "PLINK86" paranccsal azonos sorba írjuk, és megnyomjuk az Enter billentyűt:

C>PLINK86 FI [d:][\út\]<file-név1>[,<file-név2>] ...
... [,<file-név3>][LIB [d:][\út\]<könyvtár>]

ahol: "FI" és "LIB" parancsokat fent már ismertettük
"d:" a lemezegység azonosítója
"út" az elérési út
"file-név1"..."file-név3" a szerkeszteni kívánt tárgymodulok neve a .OBJ kiterjesztés nélkül
"könyvtár" a futáshoz szükséges eljárás-könyvtár neve

14.2.2.3 Interaktív üzemmód

A PLINK86 interaktív üzemmódban is képes fogadni a parancsokat a következő módon:

Gépeljük be a "PLINK86" parancsot, majd nyomjuk meg az Enter-t. Ekkor a következő prompt jelenik meg:

=>

Minden prompt után egy parancsot gépelhetünk be, amit az Enter billentyű megnyomásával kell befejezni. Ezután feltűnik az újabb prompt. Az interaktív üzemmód elhagyásához gépeljünk be egy pontosvesszőt és az Enter-t. Egy lehetséges parancssorozat:

=>FI Fopr,Pro1,Pro2
=>LIB Extend
=>;

amely a FOPR.EXE file-t eredményezi, és ez magába foglalja a PRO1.OBJ és PRO2.OBJ modulokat is.

14.2.2.4. A név.LNK file használata

Ennél a módszernél hozzunk létre egy .LNK kiterjesztésű file-t, amely tartalmazza az összes szerkesztési parancsunkat, majd gépeljük be a következő formájú parancsot:

C>PLINK86 @<file-név>

ahol: "@" azt jelzi, hogy az ezután következő file-név olyan file-t azonosít, amelyben a szükséges parancsok találhatók
"file-név" azon file neve, amely a szükséges parancsokat tartalmazza. A feltételezett kiterjesztés .LNK, ha mást nem adunk meg.

Ugyanabban a parancssorban több .LNK file is szerepelhet, pl.:

```
C>PLINK86 @Link1 @Link2
```

A PLINK86 futtatásánál is felhasználható a DOS output-irányító funkciója. Például:

```
C>PLINK86 @Link > Linkerr
```

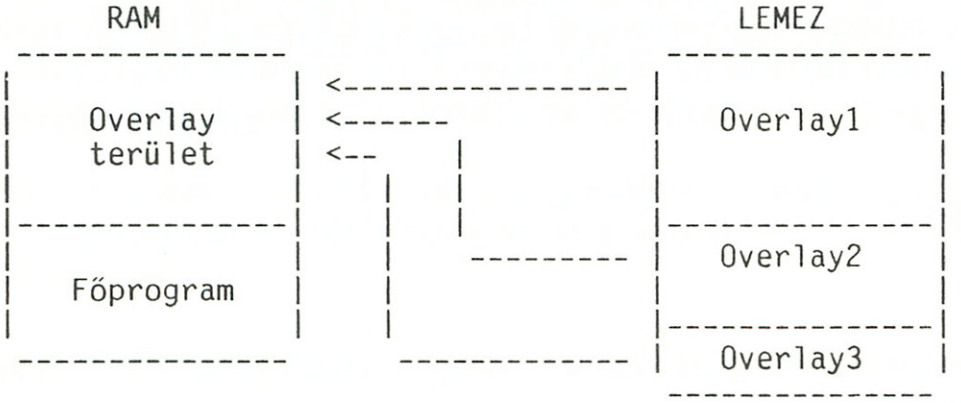
A szerkesztés után a PLINK86 a képernyőre kiírja a futtatható program nevét és méretét. Ha ehhez a mérethez hozzáadunk 64-100 kByte-t, akkor megkapjuk a program futtatásához szükséges minimális memóriaterület nagyságát.

14.2.2.5 Overlay-szervezés

Ahhoz, hogy megbizonyosodjunk afelől, hogy a felhasználói program futtatásához szükséges az overlay-k használata, próbáljuk meg előbb elindítani anélkül. A fölösleges overlay-használat nagymértékben lassítja a program végrehajtását. Az overlay-k használatára akkor kerülhet sor, ha a program mérete meghaladja a rendelkezésre álló memóriaterület nagyságát.

Az overlay-technika alkalmazásával a nagyméretű programot kisebb részekre oszthatjuk. A programrészek, vagy szegmensek maguk az overlay-k, amelyek nem tartózkodnak a RAM-ban mindaddig, amíg a végrehajtásukra sor nem kerül. A főprogram mindíg a memóriában helyezkedik el, az overlay-k viszont csak akkor töltődnek be a lemezről, mikor a főprogram meghívja azokat. Amikor a futás során egy overlay betöltésére kerül a sor, az felülírja az előzőt.

A következő ábra szemlélteti az overlay-k működését:



14.2.2.5 – 1: Az overlay-k elhelyezkedése.

Látható, hogy a legnagyobb overlay mérete határozza meg az overlay-területnek lefoglalt memória nagyságát. Igy az is világos, hogy egy overlay használata nem fogja csökkenteni a memóriaigényt.

Feltéve, hogy a program jól struktúrált, a forráskód megváltoztatása soha nem szükséges az overlay-k alkalmazásához. Az overlay-k a szerkesztés folyamán jönnek létre, és ha az overlay-struktúra megváltoztatására lenne szükség, ehhez csak a szerkesztés parancsait kell megváltoztatni, és újraszerkeszteni a programot. A forráskód ezalatt érintetlen marad. Az overlay-k megtervezése előtt fel kell osztanunk a programot olyan szegmensekre (ezek állhatnak programokból, eljárásokból, függvényekből, és

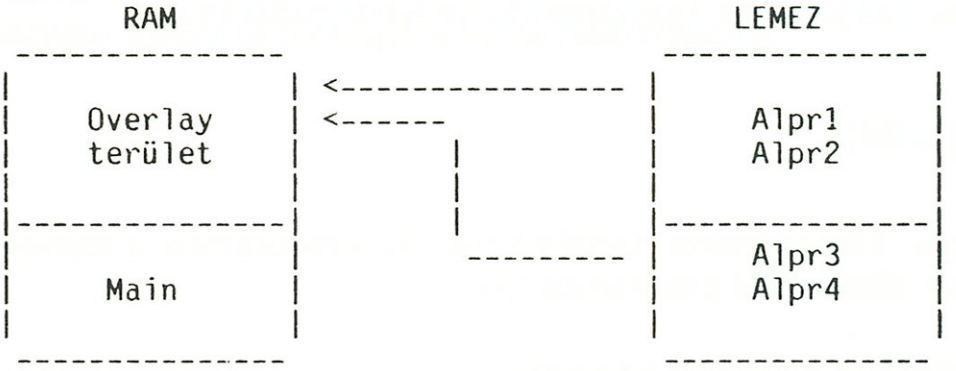
formátum- file-okból), amelyek egymástól függetlenül működnek. A főprogram lesz az, amely ezeket a logikailag elkülönített programokat meghívja.

Az overlay-k létrehozásához a .LNK file alkalmas. Az overlay-be foglalandó file-okat a "BEGINAREA" és az "ENDAREA" utasítások fogják közre. A "SECTION" és a "SECTION INTO" parancsok határozzák meg az object modulokat, amelyek az overlay-k lesznek. Az alábbi példa egy lehetséges .LNK file-t szemléltet:

```
FILE Main
LIB Extend
OVERLAY Opr,$CONSTANTS
BEGINAREA
  SECTION FILE Alpr1,Alpr2
  SECTION FILE Alpr3,Alpr4
ENDAREA
```

(A PLINK86 a szerkesztés alatt az object modulok információit osztályozza. Az adatosztály (DATA SEGMENT) az állandó információkat – pl.: string konstansok – tartalmazza, a kód-osztály (CODE SEGMENT) az aktuális programutasításokat. A Clipper a konstans-táblát általában a fő betöltő file-ba helyezi. A memóriaigény csökkentése érdekében az "OVERLAY Opr, $CONSTANTS" utasítással elérhetjük, hogy az adatosztály információi az overlay-be kerüljenek.)

A fenti példa a MAIN.EXE főprogramhoz két belső overlay-t hoz létre, amelyek két-két alprogramot fognak tartalmazni, a következő ábra szerint:



14.2.2.5 – 2: A Main program és a hozzá tartozó overlay-k működése.

A PLINK86 segítségével kétféle, belső és külső overlay hozható létre.

A belső overlay-ket a "SECTION" paranccsal határozzuk meg, és azok a .EXE file részei lesznek. A külső overlay-ket a "SECTION INTO" paranccsal hozzuk létre, és ezek önálló, .OVL kiterjesztésű file-ba kerülnek. Igy ezek olyan esetben használhatók, ha a felhasználói program például nem fér rá egy hajekony lemezre. Ilyenkor a külső overlay-k másik lemezre írhatók.

Mindazonáltal a belső overlay-k használata esetén nem szükséges külön kereső eljárást folytatni – hiszen ezek a betöltő modul végén helyezkednek el – ezért a prog-

ram végrehajtása gyorsabb lesz, mintha ugyanezt a feladatot külső overlay-kkel oldottuk volna meg.

Természetesen lehetséges több overlay-területet definiálni, sőt beágyazott overlay-ket is létrehozhatunk, de ezek a technikák már rendkívül körültekintő program- és overlaystruktúra-tervezést igényelnek.

14.3 A Clipper-t alkotó file-ok, és használatuk

A Clipper programcsomaghoz az alábbiakban felsorolt file-ok tartoznak:

14.3.1 Végrehajtható típusú (.EXE) file-ok

- CLIPPER.EXE
  A Clipper fordítót tartalmazó file
- PLINK86.EXE
  A program szerkesztőt tartalmazó file
- CREATE.EXE
  Utility a .DBF kiterjesztésű adatbázis-file-ok létrehozásához
- LABEL.EXE
  Utility a .LBL kiterjesztésű cimkeformátum-file-ok létrehozásához
- LINE.EXE
  A programokat a képernyőn, vagy nyomtatón, sorszámokkal együtt kinyomtatja
- REPORT.EXE
  Utility a .FRM kiterjesztésű jelentés szerkezeti file-ok létrehozásához

14.3.2 Könyvtárak (.LIB)

- CLIPPER.LIB
  A futáshoz szükséges könyvtárakat tartalmazza. A szerkesztés alkalmával a PLINK86 fűzi hozzá az alkalmazói programokhoz.
- OVERLAY.LIB
  Az overlay-k létrehozásához szüséges könyvtár
- EXTEND.LIB
  Bővített könyvtár

14.3.3 Egyéb kiegészítő file-ok

- READ_ME.1ST
  Egy szöveg-file, amely a Clipper-re vonatkozó legfrissebb információkat tartalmazza
- CL.BAT
  Batch file, amely megkönnyíti a file-ok fordítását és szerkesztését a fix lemezen. Egyszerű esetekben – ha nem kívánjuk a Clipper fordító opcionális parancsait, vagy a program szerkesztő parancsait alkalmazni, illetve csak a CLIPPER.LIB

könyvtárra van szükségünk – a fordítás és szerkesztés egyetlen utasítással elvégezhető:
C>CL <file-név>
ahol: "file-név" a fordítani és szerkeszteni kívánt file neve

- EXTEND*.*
  Különböző C nyelvű eljárásokat és Assembly nyelvű programokat tartalmaz a dBASE III PLUS-kompatibilitás elősegítéséhez
- FCL.BAT
  Batch file, amely – a CL.BAT-hoz hasonló módon – megkönnyíti a file-ok fordítását és szerkesztését hajlékony lemezen
- CLD.BAT
  Batch file, amely a file-ok fordítását és szerkesztését végzi fix lemezen, és hozzászerkeszti a DEBUG.OBJ file-t is
- DEBUG.OBJ
  A Clipper hibakereső programot tartalmazó object modul
- ANSI.OBJ
  A file olyan meghajtót tartalmaz, amely ANSI terminál I/O támogatással működő gépeken történő futtatáshoz szükséges
- TIPRO.PRG
  A file olyan meghajtót tartalmaz, amely TI Professionallal működő gépeken lehetővé teszi a Clipper-programok futását
- INDEX.PRG
  Utility program az index file-ok létrehozásához
- MENU.PRG
  Mintaprogram új menüparancsok használatára

14.4 Adatbáziskezelés, kompatibilitási problémák a dBASE III PLUS és a Clipper között

Azok az eltérések, amelyeket ebben a fejezetben felsorolunk, a dBASE III PLUS 1.1 változatára, és a Clipper '85 téli változatára vonatkoznak.

A parancsok és funkciók szintaxisának leírásánál – a rövidség érdekében – a továbbiakban a következő rövidítéseket alkalmazzuk:

num.kif	= numerikus kifejezés
kar.kif	= karakteres kifejezés
log.kif	= logikai kifejezés
dat.kif	= dátum kifejezés
kif.	= alfanumerikus kifejezés
kulcs kif	= az indexeléshez használt kulcskifejezés
sor	= a képernyő, vagy nyomtató sorának száma (Lehet aritmetikai kifejezés is, amely memóriaváltozót is tartalmazhat.)
oszlop	= a képernyő, vagy nyomtató oszlopának száma (Lehet aritmetikai kifejezés is, amely memóriaváltozót is tartalmazhat.)
mem.vált	= memóriaváltozó
file-név	= a file neve kiterjesztés nélkül
file-azonosító	= a file neve kiterjesztéssel együtt
elj.név	= az eljárás neve
alias	= a file hivatkozási neve
érv.kör	= a műveletek érvényességi köre
d:	= a meghajtó azonosítója
út	= a file elérési útja
munkater	= munkaterület
param	= paraméter
skeleton	= egy olyan file-azonosító, amelyben legalább egy behelyettesítendő karakter szerepel, azaz "*" vagy "?" vagy mindkettő.

14.4.1 A Clipper által nem támogatott dBASE III PLUS funkciók és parancsok

A Clipper nem támogatja azokat a dBASE III PLUS parancsokat, amelyek elsősorban interaktív üzemmódban használatosak. Ezen parancsok hiányából adódó nehézségek általában áthidalhatók a különböző Clipper Utility programok segítségével. (Részletesebb információk a Clipper Utility-kről a 14.4.4. fejezetben találhatók.)

A Clipper által nem támogatott dBASE III PLUS parancsok és funkciók felsorolása:

APPEND
ASSIST
BROWSE
CHANGE
CLEAR FIELDS
CREATE LABEL
CREATE REPORT
CREATE QUERY
CREATE SCREEN
CREATE VIEW
DISPLAY FILES
DISPLAY MEMORY
DISPLAY STATUS
DISPLAY STRUCTURE
DISPLAY USERS
EDIT
ERROR()
EXPORT TO
HELP
IMPORT FROM
INSERT
LIST FILES
LIST HISTORY
LIST MEMORY
LIST STATUS
LIST STRUCTURE
LOAD
LOGOUT
MESSAGE
MODIFY COMMAND
MODIFY LABEL
MODIFY QUERY
MODIFY REPORT
MODIFY SCREEN
MODIFY STRUCTURE
MODIFY VIEW
ON ERROR/ESCAPE/KEY
RESUME
RETRY
RETURN TO MASTER
SET
SET CARRY
SET CATALOG
SET COLOR ON/OFF
SET DEBUG
SET DOHISTORY
SET ECHO
SET ENCRYPTION
SET FIELDS
SET HEADING
SET HELP
SET HISTORY
SET MEMOWITH
SET MENUS
SET SAFETY
SET STATUS
SET HELP
SET TALK
SET TITLE
SET TYPEAHEAD
SET VIEW

Vannak olyan parancsok és funkciók, amelyek a dBASE III PLUS-ban, és a Clipperben is léteznek, de a Clipper eltérő módon támogatja azokat. Ezeket a következő fejezetben ismertetjük.

14.4.2 A Clipper újdonságai a dBASE III PLUS-hoz képest

Ebben a fejezetben azokat a Clipper-parancsokat és funkciókat ismertetjük, amelyek a dBASE III PLUS-ban egyáltalán nem, vagy eltérő használattal léteznek.

Parancsok:

14.4.2.1 @...BOX

Szintaxis: @ <sor1>,<oszlop1>,<sor2>,<oszlop2> BOX <kar.kif>
ahol: "sor1" a felső sor (0-24 közötti érték).
"oszlop1" a baloldali oszlop (0-79 közötti érték).
"sor2" az alsó sor (0-24 közötti érték).
"oszlop2" a jobboldali oszlop (0-79 közötti érték).
"kar.kif" egy karakter-sorozat, amely meghatározza az oldalakat és a sarkokat, valamint egy kitöltő karakter. A sztring-ben megadott karakterek a bal felső saroktól kezdve, az óramutató járásával egyező irányban jelennek meg.
Cél: Négyszöget rajzol a képernyőre, amelynek méretét, elhelyezkedését, és az azt alkotó grafikus karaktereket meghatározhatjuk.
Példa: olyan négyszöget rajzol, amelynek vizszintes oldalait dupla, függőleges oldalait szimpla vonal alkotja. Kitöltő karaktert nem alkalmazunk.

```
dsz = CHR(213) + CHR(205) + CHR(184) + CHR(179) + CHR(190) + ;
CHR(205) + CHR(212) + CHR(179)
@ 5,10,20,70 BOX dsz
```

Könyvtár: CLIPPER.LIB

14.4.2.2 @...SAY...GET

Szintaxis: @ <sor>,<oszlop> [SAY <kif> [PICTURE <kar.kif>]]
[GET <mem.vált> [PICTURE <kar.kif>]
[RANGE <num.kif1>,<num.kif2>][VALID <log.kif>]]
Eltérés: A parancs bővült a "VALID" opcióval, amely a "RANGE"-hez hasonlóan engedélyezi, vagy letiltja az input elfogadását. Megjegyzendő, hogy "log.kif" tartalmazhat a felhasználó által definiált függvényt.
Példa:

```
a = 0
@ 10,1 SAY "Kerek egy pozitiv szamot: " GET a;
VALID a > 0
```

Könyvtár:CLIPPER.LIB

14.4.2.3 APPEND FROM

Szintaxis: APPEND [<érv.kör>][FIELDS <mezőlista>] FROM
<file-név> [FOR <feltétel>][WHILE <feltétel>]
[SDF]/[DELIMITED[WITH BLANK/<határoló>/(<kar.kif>)]]
Eltérés: A parancs a "FIELDS" opcionális argumentummal bővült. Ha a mezőlistát megadjuk, az adatok csak ezekbe a mezőkbe töltődnek be.
Példa: a lakás-telefonnal rendelkező dolgozók nevét és címét másolja át a telefonkönyv file-ba

```
USE Phone
APPEND FIELDS nev, cim FROM Dolgozok FOR otel > " "
```

Könyvtár: CLIPPER.LIB

14.4.2.4 AVERAGE

Szintaxis: AVERAGE [<érv.kör>] <mezőlista> TO <mem.vált lista>
[FOR <feltétel>][WHILE <feltétel>]
Eltérés: A mezőlista és a memóriaváltozó-lista kitöltése kötelező.
Példa: Az 1988-ban lekönyvelt bevételeket és kiadásokat átlagolja

```
USE Konyvel
AVERAGE FIELDS bevetel,kiadas TO atlb,atlk FOR ev = 1988
```

Könyvtár: CLIPPER.LIB

14.4.2.5 CALL

Szintaxis: CALL <elj.név> [WITH <param. lista>]
ahol: "elj.név" a végrehajtandó eljárás neve
"WITH" meghatározza az átadandó paramétereket
"param. lista" a paraméterek sorozata vesszővel elválasztva
Cél: A "CALL" paranccsal hívható eljárások lehetnek Clipper-ben, C vagy Assembly nyelven írottak. Csak az a követelmény, hogy lefordítva, .OBJ kiterjesztésű modulként álljanak a PLINK86 rendelkezésére. Ha az eljárást magasabb szintű nyelven írtuk, akkor az illető "run-time"-, és szerkesztő könyvtárnak is elérhetőnek kell lennie a program szerkesztő számára. A paraméter-lista legfeljebb 7 db. paraméterből állhat. Valamennyi adat hivatkozásként adódik át.

Könyvtár: CLIPPER.LIB

14.4.2.6 COUNT

Szintaxis: COUNT [<érv.kör>][FOR <feltétel>][WHILE <feltétel>]
TO <mem.vált>
Eltérés: A memóriaváltozó megadása kötelező.
Példa: Megszámlálja az 5000Ft-nál kisebb fizetésü dolgozókat

```
USE Dolgozok
COUNT FOR ber < 5000 TO szegeny
? szegeny
```

Könyvtár: CLIPPER.LIB

14.4.2.7 CREATE

Szintaxis: CREATE <file-név>
Eltérés: Bővített szerkezetű file-t hoz létre. Új adatbázis kialakításához a "CREATE FROM" paranccsal együtt használva alkalmas.

A bővitett szerkezetű file felépítése:

mezőnév	típus	hossz
Field_name	C	10
Field_type	C	1
Field_len	N	3
Field_dec	N	3

Példa:

```
CREATE Stru
DO WHILE .T.
   mezonev = SPACE(10)
   mezotip = SPACE(1)
   mezohossz = 0
   mezotized = 0
   CLEAR
   @ 5,2 SAY "Nev: " GET mezonev
   @ 7,2 SAY "Tipus: " GET mezotip
   @ 9,2 SAY "Hossz: " GET mezohossz
   @ 11,2 SAY "Tizedes: " GET mezotized
   READ
   IF LEN(TRIM(mezonev)) < > 0
     APPEND BLANK
     REPLACE Field_name WITH mezonev
     REPLACE Field_type WITH mezotip
     REPLACE Field_len WITH mezohossz
     REPLACE Field_dec WITH mezotized
   ELSE
     USE
     EXIT
   ENDIF
 ENDDO
 CREATE Adatb FROM Stru
```

Könyvtár: CLIPPER.LIB

14.4.2.8 DECLARE

Szintaxis: DECLARE <tömb1>[<num.kif1>],<tömb2>[<num.kif2>]...
ahol: "tömb1","tömb2" a létrehozni kivánt tömb neve
"num.kif1","num.kif2" a tömbök elemeinek száma
A szögletes zárójelek ([]) itt a parancs részei, nem opcionális argumentumot jelölnek!
Cél: Ezzel az utasítással egydimenziós tömböket határozhatunk meg. A tömb elemei kevert típusúak is lehetnek. Az elemek számának felső korlátja 2048. A tömbök mindíg "PRIVATE" változók lesznek, és nem deklarálhatók "PUBLIC"-ként. A tömbök és elemeik nem tárolhatók el .MEM file-okba, és nem adhatók át hivatkozási paraméterként a "DO" és "CALL" parancsokkal.

A "LEN()" függvény – ha a tömb neve a paraméter – az elemek számával tér vissza. A "TYPE()" függvény – ha a tömb neve a paraméter – mindíg "A"-val tér vissza. A tömbök deklarálásánál a zárójelek nem lehetnek makro-változók elemei.
Példa:

```
tomb = "szavak"
szam = 3
DECLARE &tomb[szam]
&tomb[1] = "abc"
szavak[3] = .T.
STORE 12 TO &tomb[2]
FOR i = 1 TO 3
   ? szavak[i]
NEXT
```

Eredmény: abc
12
.T.

Könyvtár: CLIPPER.LIB

14.4.2.9 DISPLAY

Szintaxis: DISPLAY [OFF][< érv.kör >] < mezőlista > [FOR < feltétel >]
[WHILE < feltétel >]
Eltérés: A mezőlista kötelezően kitöltendő.
Példa: Kiírja a dolgozók nevét és címét

```
USE Dolgozok
DISPLAY nev,cim
```

Könyvtár: CLIPPER.LIB

14.4.2.10 EXTERNAL

Szintaxis: EXTERNAL < elj.név lista >
ahol: "elj.név lista" az eljárások, felhasználói funkciók,
és formátum file-ok listája, amelyeket a szimbólum-
táblázatba akarunk foglalni.
Cél: Szimbólumok meghatározása a program szerkesztő számára. Ha az eljárásokat, vagy felhasználói függvényeket makróval hívjuk, vagy overlay-be akarjuk helyezni, akkor "EXTERNAL"-ként kell definiálni.
Példa:

```
EXTERNAL Elj1,Elj2,Elj3
szam = 2
DO Elj&szam
```

Könyvtár: CLIPPER.LIB

14.4.2.11 FOR...NEXT

Szintaxis: FOR <mem.vált> = <num.kif1> TO <num.kif2> [STEP <num.kif3>]
<parancsok>
[EXIT]
<parancsok>
NEXT
ahol: "mem.vált" a kontrolváltozó
"num.kif1" a kontrolváltozó kezdeti értéke
"num.kif2" a kontrolváltozó felső vagy alsó korlátja
"STEP" parancs változtatja a kontrolváltozó értékét
"num.kif3" a változás mértéke – lehet negatív is (Ha nem használjuk a "STEP" parancsot, akkor az alapértelmezés a változás mértékére 1.)
Cél: A kifejezések egy tartományán belül ciklikus műveleteket hajt végre. A kontrolváltozó és a változás mértéke dinamikusak, azaz ciklus közben változtathatók.
Példa: a legjobban tömbökkel végzett műveleteknél használhatók:

```
hossz = LEN(tomb)
FOR i = 1 to hossz
        tomb[i] = i*3
NEXT
```

Könyvtár: CLIPPER.LIB

14.4.2.12 FUNCTION...RETURN

Szintaxis: FUNCTION <elj.név>...RETURN <kif>
ahol: "elj.név" a felhasználói függvény neve (10 karakter hosszúságú lehet)
"kif" az a mennyiség, amellyel a függvény visszatér
Cél: A felhasználó által definiált függvényeket határozza meg. Hívásuk a Clipper függvényeihez hasonlóan történik: eljárás-név (<param. lista>)
Példa: így használható a "VALID" opció után

```
szám = 0
@ 10,10 SAY "Kerem a szamot: " GET szam VALID erv(szam)
READ
...
FUNCTION erv
PARAMETER szam
RETURN (szam > 0)
```

Könyvtár: CLIPPER.LIB

14.4.2.13 KEYBOARD

Szintaxis: KEYBOARD <kar.kif>
ahol: "kar.kif" az a sztring, amivel a billentyűzet pufferét feltöltjük.

Cél: Nagyon hasznos minden olyan parancs, vagy funkció alkalmazásakor, amely a billentyűzetről várja az inputot. Minden "KEYBOARD" parancs törli a billentyűzet puffert, így a puffer kiürítésére a KEYBOARD "" alkalmas.
Példa: A "RETURN TO MASTER" szimulálása olyan menürendszerben, ahol minden menüből "Q"-val lehet kilépni

```
KEYBOARD "Q" + CHR(13) + ... + "Q" + CHR(13)
```

Könyvtár: CLIPPER.LIB

14.4.2.14 LIST

Szintaxis: LIST [OFF] [<érv.kör>] <mezőlista> [FOR <feltétel>]
[WHILE <feltétel>] [TO PRINT/TO FILE <file-név>]
Eltérés: A mezőlistát kötelező kitölteni, és új opciók a "TO PRINT" és a "TO FILE".
"TO PRINT" a printerre irányítja az outputot
"TO FILE <file-név>" a "file-név" azonosítójú file-ba irányítja az outputot. Ha más kiterjesztés nincs megadva, akkor az alapértelmezés .TXT.
Példa: Printerre írja a dolgozók nevét és címét

```
USE Dolgozok
LIST nev,cim TO PRINT
```

Könyvtár: CLIPPER.LIB

14.4.2.15 PRIVATE

Szintaxis: PRIVATE <mem.vált lista>
Eltérés: A memóriaváltozók listája kötelezően kitöltendő.
Példa:

```
PRIVATE b
? TYPE ("a")     Eredmény: U
? TYPE ("b")               U
DO valt
? TYPE ("a")               U
? TYPE ("b")               C
...
PROCEDURE valt
        a = "a"
        b = "b"
RETURN
```

Könyvtár: CLIPPER.LIB

14.4.2.16 PROCEDURE

Szintaxis: PROCEDURE <elj.név> ... RETURN
Eltérés: Az eljárások a főprogram végére illeszthetők, de ilyenkor egyedi névvel kell rendelkezniük.

Példa:

```
* Fopr.prg
DO Elj
RETURN
PROCEDURE Elj
        ? "Szia !"
RETURN
```

Könyvtár: CLIPPER.LIB

14.4.2.17 RESTORE SCREEN

Szintaxis: RESTORE SCREEN
Cél: A képernyő visszatöltésére szolgál. A "SAVE SCREEN" paranccsal együtt használatos, és segítségével elkerülhető a képernyő újrarajzolása egy hibaüzenet, vagy help-képernyő behívása után.

Könyvtár: CLIPPER.LIB

14.4.2.18 SAVE SCREEN

Szintaxis: SAVE SCREEN
Cél: A képernyő elmentésére szolgál. A "RESTORE SCREEN" paranccsal együtt használatos, és segítségével elkerülhető a képernyő újrarajzolása egy hibaüzenet, vagy help-képernyő behívása után.
Példa:

```
USE Cipo
SEEK "42"
IF EOF()
        SAVE SCREEN
        @ 10,25,20,55 BOX doboz
        @ 15,29 SAY "Nincs ilyen nagysagu cipo."
        INKEY()
        RESTORE SCREEN
ENDIF
```

Könyvtár: CLIPPER.LIB

14.4.2.19 SET COLOR TO

Szintaxis: SET COLOR TO [<normál> [,<kiemelt>] [,<keret>]]
Eltérés: A normál, kiemelt és keret-színek beállítására makrok is használhatók, és ebben az egyedi esetben a makro-változóban vesszők is megengedettek.
Példa:

```
szin = "W/R,R/W"
SET COLR TO &szin
@ 10,10 SAY "Indulhat a nyomtatás? (I/N): " GET val
```

READ
SET COLOR TO

Könyvtár: CLIPPER.LIB

14.4.2.20 SET ESCAPE

Szintaxis: SET ESCAPE ON/off
Eltérés: Meghatározza, hogy megszakítsa-e az Alt-C a program végrehajtását, illetve az Esc a "READ" parancs végrehajtását.
Ha a "SET ESCAPE ON" érvényes, akkor az Alt-C a "(Q/A/I)" üzenetet küldi, ahol
Q = kilépés
A = megszakítás, és visszatérés a DOS-hoz
I = a hibát figyelmen kívül hagyja, és folytatás

Könyvtár: CLIPPER.LIB

14.4.2.21 SET FUNCTION

Szintaxis: SET FUNCTION <num.kif> TO <kar.kif>
Eltérés: A "SET FUNCTION 1" a Help.prg nevű file-nak van fenntartva, és nem változtatható meg. A Help.prg nevű file viszont minden programhoz az igényeknek megfelelően állítható össze, így segítséget nyújt a felhasználó által definiált help-képernyők használatához.
A "SET KEY" parancs ugyanarra a funkcióbillentyűre precedenciát élvez a "SET FUNCTION" paranccsal szemben.

Könyvtár: CLIPPER.LIB

14.4.2.22 SET KEY

Szintaxis: SET KEY <num.kif> TO [<elj.név>]
ahol: "num.kif" a kérdéses billentyű ASCII kódja.
"elj.név" az eljárás neve, amely a billentyű lenyomására végrehajtódik.
Cél: Egy eljárás végrehajtása egyetlen billentyű lenyomására bármely várakozási állapotból. A várakozási állapotot olyan parancsok eredményezik, amelyek szüneteltetik a program végrehajtását:
WAIT
READ
ACCEPT
INPUT
MENU TO
Legfeljebb 32 db "SET KEY" parancs lehet érvényben egyidőben. Az F1 billentyű a Help.prg nevű file-nak van fenntartva, és nem változtatható meg. A paranccsal automatikusan három paraméter adódik át:

- a programnév
- a sor száma
- az input-változó

Példa: Ctrl-F1 hatására kiírja az időt

```
SET KEY -20 TO Ido
PROCEDURE Ido
PARAMETERS a,b,c
SAVE SCREEN
@ 0,0 CLEAR
@ 10,10 SAY TIME()
INKEY()
RESTORE SCREEN
RETURN
```

Könyvtár: CLIPPER.LIB

14.4.2.23 SET MESSAGE TO...@...PROMPT...MENU TO

Szintaxis: SET MESSAGE TO [sor1]
@ <sor2>,<oszlop> PROMPT <kar.kif1> [MESSAGE <kar.kif2>]
MENU TO <mem.vált>

ahol: "sor1" az a sorszám, ahol az üzenet megjelenik
"sor2" az a sorszám, ahol a prompt megjelenik
"oszlop" az oszlop száma, ahol a prompt megjelenik
"kar.kif1" a prompt szövege
"kar.kif2" az üzenet szövege
"mem.vált" a választás eltárolására szolgál

Cél: Menük egyszerű létrehozását segíti elő. A "@...PROMPT" paranccsal a menü pontjait helyezhetjük el a képernyőn, és köztük a kurzormozgató billentyűk segítségével mozoghatunk. Ilyenkor minden menüponthoz hozzárendelhető egy üzenet, amely a "SET MESSAGE" parancsban meghatározott sorban fog megjelenni. Minden menühöz maximum 32 pont tartozhat. A "MENU TO" utasítás a – "READ"-hez hasonlóan működik – a választást eltárolja a "mem.vált" numerikus változóba, amely a kiválasztott prompt viszonylagos sorszáma lesz. (Az első menüpont választásához az 1 tartozik.) Menüzés közben hívhatók a "SET KEY" paranccsal meghatározott eljárások is, de ha ezekben nem "PRIVAT"-ként deklaráltuk a paramétereket, akkor az eljáráson belül felülíródhat a választást tartalmazó memóriaváltozó. A menük egymásba ágyazhatók, azonban célszerű minden menühöz egyedi memóriaváltozót kijelölni a választás megőrzésére. Ha Esc beütésével hagyjuk el a menüt, a választás értéke 0 lesz.
Példa:

```
CLEAR
SET MESSAGE TO 23
@ 1,5 PROMPT "Torles" MESSAGE "A teljes adatbazis torlese"
@ 2,5 PROMPT "Modositas" MESSAGE "A tetelek torlese;
vagy uj tetelek felvitele"
@ 3,5 PROMPT "Listazas" MESSAGE "A teljes adatbazis;
kiirasa printerre"
MENU TO val
```

Könyvtár: CLIPPER.LIB

14.4.2.24 SET RELATION TO

Szintaxis: SET RELATION TO [<kulcs kif1>/<RECNO()>/<num.kif1>
INTO <alias1>][,TO <kulcs kif2>/<RECNO()>/<num.kif2>
INTO <alias2>]...
Eltérés: Többszörös 'szülő-gyerek'-kapcsolatok is megengedettek, a 'gyerek'-file-ok száma egy 'szülő'-höz legfeljebb 8 lehet.
Példa:

```
SELECT 1
USE Dolgozok INDEX nev ALIAS dolg
SELECT 2
USE Uzorv INDEX mhely ALIAS uz
SELECT 3
USE Kororv INDEX lakhely ALIAS kor
SELECT 1
SET RELATION TO mhely INTO uz, TO lakhely INTO kor
```

Könyvtár: CLIPPER.LIB

14.4.2.25 SUM

Szintaxis: SUM [<érv.kör>]<kif. lista> TO <mem.vált lista>
[FOR<feltétel>][WHILE <feltétel>]
Eltérés: A memóriaváltozó-lista kitöltése kötelező, a kifejezéslista és memóriaváltozó-lista elemeinek száma azonos kell hogy legyen.
Példa:

```
USE Konyveles
SUM bevetel-30 TO osszeg
```

Könyvtár: CLIPPER.LIB

14.4.2.26 &&

Szintaxis: NOTE/* [<szöveg>]/[<parancs>] && [<szöveg>]
ahol: "szöveg" az a karaktersorozat, amit kommentárként kívánunk a programba írni.
Cél: "&&" a sorban bárhol feltüntethető, ettől kezdve a Clipper nem értelmezi a sort, amíg egy sorvége-karaktert nem talál. Igy megjegyzések helyezhetők el a parancsokkal azonos sorban is.
Példa:

```
* Ha a valasztas = nem, figyelmeztetes
IF val = "n"
        SAVE SCREEN   && Elmentjuk a kepernyot
        @ 10,10 SAY uzenet
        ...
ENDIF
```

Könyvtár: CLIPPER.LIB

Funkciók:

14.4.2.27 DTOS()

Szintaxis: DTOS(<dat.kif>)
ahol: "dat.kif" egy dátum-tipusú változó
Cél: Egy dátumot olyan karakterlánccá alakít, amely alkalmas index-kulcsnak. A "DTOS()" egy 8 karakter hosszúságú, "yyyymmdd" formátumú karaktersorozattal tér vissza.
Példa:

```
a = DATE()
b = DTOS(DATE())
?? TYPE("a")
?? TYPE("b")
            Eredmény: D C
```

Könyvtár: CLIPPER.LIB

14.4.2.28 EMPTY()

Szintaxis: EMPTY(<kif>)
ahol: "kif" bármilyen tipusú kifejezés lehet
Cél: Az "EMPTY()" .T.-val tér vissza, a következő táblázat szerint:

Adattípus	Tartalom
Karakteres	üres, vagy szóközök
Numerikus	0
Dátum	üres
Logikai	.F.

Példa:

```
? EMPTY("")              Eredmény: .T.
? EMPTY(SPACE(5))        Eredmény: .T.
```

Könyvtár: CLIPPER.LIB

14.4.2.29 FIELDNAME()

Szintaxis: FIELDNAME(<num.kif>)
ahol: "num.kif" a mező sorszáma az aktív adatbázisban
Cél: Az adatbázisból kiválasztott mező nevével tér vissza. Ha "num.kif" 0, vagy nagyobb a mezők számánál, akkor üres sztring lesz az eredmény. Egyébként a mezőnevek nagybetűvel térnek vissza.
Példa:

```
USE Dolgozók
? FIELD(40)
      Eredmény: 0
```

Könyvtár: CLIPPER.LIB

14.4.2.30 IF()

Szintaxis: IF(<log.kif>,<kif1>,<kif2>)
ahol: "log.kif" logikai kifejezés
"kif1" a visszatérő érték, ha "log.kif" igaz
"kif2" a visszatérő érték, ha "log.kif" hamis
Cél: A logikai kifejezés értékétől függő kifejezéssel tér vissza. "kif1" és "kif2" nem lehetnek különböző tipusúak.
Példa:

```
a = 50
b = IF(a < 10,"kisebb","nem kisebb")
                                    Eredmény: nem kisebb
```

Könyvtár: CLIPPER.LIB

14.4.2.31 INKEY()

Szintaxis: INKEY([<num.kif>])
ahol: "num.kif" meghatározza, hogy hány másodpercig várakozzon a billentyű lenyomására. Ha "num.kif" értéke 0, akkor mindíg megvárja a billentyű leütését.
Cél: Beolvas egy karaktert a billentyűzetről. Egy numerikus értékkel tér vissza, amely megegyezik a billentyű ASCII kódjával (azaz az értékkészlet -39 és 386 közé esik)
Példa: Esc hatására nem kezdi meg a nyomtatást

```
? "Allitsd a papirt laptetore!"
? "Ha kesz, nyomj meg egy gombot!"
IF INKEY() = 27
        RETURN
ENDIF
```

Könyvtár: CLIPPER.LIB

14.4.2.32 LASTKEY()

Szintaxis: LASTKEY()
Cél: Meghatározza a várakozó helyzetben utoljára megnyomott billentyűt. A billentyű ASCII kódjának értékével tér vissza (-39 és 366 között). Jól használható annak ellenőrzésére, hogy hogyan fejeződött be a "READ".
Példa: Esc hatására nem ír az adatbázisba

```
USE Dolgozok
@ 10,10 SAY "Uj lakcim: " GET cim
READ
IF LASTKEY() < > 27
        REPLACE lakcim WITH cim
ENDIF
USE
```

Könyvtár: CLIPPER.LIB

14.4.2.33 LASTREC()

Szintaxis: LASTREC()
Cél: A fizikai rekordok számát határozza meg az aktuális adatbázisban. Egy numerikus értékkel tér vissza. A "SET FILTER" illetve "SET DELETED" parancsok nincsenek hatással a visszatérő értékre.
Példa:

```
USE Dolgozok
? LASTREC()              Eredmény: 200
SET FILTER TO nev = "Kovacs"
? LASTREC()              Eredmény: 200
```

Könyvtár: CLIPPER.LIB

14.4.2.34 MEMOEDIT()

Szintaxis: MEMOEDIT(<kar.kif> [,<sor1>,<oszlop1>,<sor2>,
<oszlop2>][,<log.kif>])
ahol: "kar.kif" a karakterlánc, vagy mező, amit szerkeszteni akarunk
"sor1","oszlop1","sor2","oszlop2" a szerkesztő ablak koordinátái a következő sorrendben: felső, bal, alsó, jobb.
"log.kif" meghatározza, hogy a memo szerkeszthető (.T.), vagy csak megtekinthető (.F.). Az alapértelmezés .T..
Cél: Karaktersztringek illetve mezők kiíratása, vagy szerkesztése. A memo megfelelő helyeinek kikeresésére a kurzormozgató billentyűket használjuk.

Szerkesztő billentyűk:

Billentyű	Szerep
Ctrl-Y	törli a sort
Ctrl-T	törli a jobbra eső szót
Ctrl-B	újraalakítja a memo-t a szerkesztő ablakban
Ctrl-W	a szerkesztés befejezése mentéssel
Esc	a szerkesztés megszakítása

Példa: memo szerkesztése

```
REPLACE abc WITH MEMOEDIT(abc,10,30,20,50,.T.)
```

Könyvtár: EXTEND.LIB

14.4.2.35 PROCLINE()

Szintaxis: PROCLINE()
Cél: Az éppen futó program forráskódjának aktuális sorával tér vissza. Ha a fordításkor a "-l" opcióval letiltottuk a sorszámok eltárolását, akkor a visszatérő érték előreláthatatlan lesz!

Példa: hibakeresési, vagy ellenőrzési céllal

```
? PROCLINE(), a,b,c
```

Könyvtár: CLIPPER.LIB

14.4.2.36 PROCNAME()

Szintaxis: PROCNAME()
Cél: Az éppen futó eljárás nevét írja ki.
Példa:

```
? "Az " + PROCNAME() + " eljaras folyik"
```

Könyvtár: CLIPPER.LIB

14.4.2.37 REPLICATE()

Szintaxis: REPLICATE(<kar.kif>,<num.kif>)
ahol: "kar.kif" az ismételendő karaktersorozat
"num.kif" az ismétlés száma
Cél: A "REPLICATE()" függvénnyel egy karaktersztring meghatározott számban ismételhető. A visszatérő karaktersztring maximális hossza 64 kByte lehet. Ha "num.kif" értéke 0, akkor üres sztring tér vissza.
Példa: csillagokból álló sor rajzolása

```
? REPLICATE("*",79)
```

Könyvtár: CLIPPER.LIB

14.4.2.38 SECONDS()

Szintaxis: SECONDS()
Cél: Az éjfél óta eltelt idővel tér vissza másodpercekben. Igy az értékkészlet 0-86399.
A visszatérő numerikus érték formátuma: "másodpercek.századok"
Példa: egy eljárás futtatási idejének kiíratása

```
a = SECONDS()
DO Eljaras
b = SECONDS()
? "A futasi ido: " + STR(b-a) + " masodperc"
```

Könyvtár: CLIPPER.LIB

14.4.2.39 SELECT()

Szintaxis: SELECT()
Cél: Egy egész számmal tér vissza, amely az éppen aktuális adatbázis számát mutatja (0-254)

Példa:

```
SELECT 1
? SELECT()        Eredmény: 1
```

Könyvtár: CLIPPER.LIB

14.4.2.40 UPDATED()

Szintaxis: UPDATED()
Cél: Egy logikai értékkel tér vissza. Az "UPDATED" igaz lesz, ha az utolsó, vagy aktuális "READ" megváltoztatta a "GET"-ek valamely adatát.
Példa:

```
USE Dolgozok
@ 10,10 GET vnev
READ
IF UPDATED()
  REPLACE nev WITH vnev
ENDIF
```

Könyvtár: CLIPPER.LIB

14.4.3 Működési különbségek

14.4.3.1 Memóriamezők fejlettebb kezelése

A memóriamezők sztring értékként tárolhatók a memória-változókban, így lehetőség nyílik bármilyen sztring-művelet elvégzésére rajtuk.

14.4.3.2 A makro-szerkezetek új lehetőségei

A Clipper megengedi makro-k használatát a "DO WHILE" ciklusokban, valamint lehetővé teszi rekurzív makro-k használatát is.
Ha a makro-kat parancssorokban használjuk, tartsuk szem előtt, hogy a program fordításához szükséges minden szintaktikus elem jelenléte, igy a makro-k nem tartalmazhatnak parancsokat, vagy parancs-vesszőket. (Vesszők használata makro-kban kizárólag a "SET COLOR" paranccsal engedhető meg!)

14.4.3.3 Uj operátorok

= = (Exactly equal)
Karakteres kifejezések összehasonlításánál teljes egyezőséget kíván – azonos hossz, azonos tartalom. Más típusokra (numerikus, logikai, stb.) működése megegyezik a " = "-évél, de csak a 12 legnagyobb helyiérték összehasonlítása történik meg.

! (Not)
Működése megegyezik a .NOT. operátoréval.

% (Modulus)
A modulus matematikai operátor az osztás maradékával tér vissza.

14.4.3.4 Index file-ok

A Clipper által generált index file-ok kiterjesztése – a dBASE III PLUS .NDX file-jaival szemben – .NTX . Az .NTX file-ok működése észrevehetően gyorsabb, és méretük is kisebb az .NDX file-okénál.

14.4.3.5 Kurzorvezérlő billentyűk

A "GET" parancsok alkalmazásakor a képernyő szerkesztéséhez a következő billentyűk használhatók:

Billentyű		Hatás
Leftarrow	Ctrl-S	Egy karakterrel balra lép, de nem lép át az előző "GET"-be
Rightarrow	Ctrl-D	Egy karakterrel balra lép. A "GET" végén a kurzor átlép a következő "GET"-be
Ctrl-Leftarrow	Ctrl-A	Egy szóval balra lép
Ctrl-Rightarrow	Ctrl-F	A következő szó elejére lép
Uparrow	Ctrl-E	Előző "GET"-re lép
Dnarrow	Ctrl-X	Következő "GET"-re lép
Home		Az aktuális "GET" elejére lép
End		Az aktuális "GET" utolsó karakterére lép
Ctrl-Home		Az előző "GET"-re lép
Ctrl-End		A következő "GET"-re lép
Del	Ctrl-G	Törli a kurzorpozíción lévő karaktert
Backspace	Ctrl-H	Törli a kurzorpozíciótól balra eső karaktert
Ctrl-T		Törli a kurzortól jobbra eső szót
Ctrl-Y		Törli a karaktereket a kurzorpozíciótól a "GET" végéig.
Ins	Ctrl-V	Insert mód
Ctrl-W		Befejezi a "READ" parancsot, és elmenti a "GET" értékét
Ctrl-C		
PgUp		
PgDn		
Esc		Befejezi a "READ" parancsot, de nem menti el a "GET" értékét
Ctrl-U		Visszatölti a "GET" eredeti értékét

14.4.4. Nyomkövetés, tesztelés (a Clipper Debugger)

A Clipper Debugger segítségével könnyen felderíthetők a programhibák, annak köszönhetően, hogy a nyomkövető a következő szolgáltatásokat nyújtja:

- A program végrehajtható normális sebességgel, vagy lépésről-lépésre.
- A programkörnyezet állapota ("SET" parancsok) kiíratható

- Bármely kifejezés értéke megtekinthető.
- Bármely memóriaváltozó tartalma ellenőrizhető, illetve megváltoztatható.
- Definiálhatunk megszakítási pontokat, amelyeknél a program megállítását kívánjuk.

A Clipper Debuggert a DEBUG.OBJ file tartalmazza. Ezt kell hozzászerkeszteni az alkalmazói programhoz, ha a nyomkövetést akarjuk használni. A DEBUG.OBJ akármelyik könyvtárban elhelyezkedhet, ha a DOS-ból elérhető ez a könyvtár.

Egy lehetséges parancssor látható alább a hibakereső alkalmazására:

```
C>PLINK86 FI Fopr,\CLIPPER\DEBUG
```

(Ne felejtsük el, hogy a CLD.BAT batch file automatikusan elvégzi a hibakereső hozzáfűzését a programunkhoz!)

Lényeges az is, hogy a szerkesztésnél elsőként a felhasználói program tárgymodulját adjuk meg, ellenkező esetben a .EXE file nem fog működni.

– A nyomkövető használata:

Ha programjainkat a nyomkövetővel futtatjuk, a DOS betölti a programot a memóriába, és a képernyő felső sorában a következő sor jelenik meg:

```
proc:    line:    ([F1]-help,B,C,D,E,G,L,M,P,Q,R,S,T,V):
```

a "proc:" után a program, vagy eljárás neve áll
a "line:" után a következő végrehajtandó sor száma áll

a zárójelben lévő karakterek pedig a hibakeresőnek adható parancsokat azonosítják, amelyet az utolsó kettőspont után írhatunk be.

A kurzor mindíg a nyomkövető parancssorában marad, kivéve ha a program futása során adat bevitelére van szükség. Ennek befejezése után a kurzor ismét a parancssorba tér vissza.

– A nyomkövető parancsai:

Alt-D a program végrehajtását megállítja, és megjelenik a hibakereső parancssora.

F1 hatására megjelenik a help-képernyő.

B (Breakpoint) Megszakítási pont megadására alkalmas. Legfeljebb 9 db megszakítási pontot adhatunk meg a sorszám meghatározásával, és legfeljebb egyet feltétel megadásával. A "B" parancs hatására a következő sor jelenik meg: Break number (0, 1-9):
Ha erre Enter-t ütünk be, megnézhetjük a jelenlegi megszakítási pontok helyét. Ha 0-t adunk meg, akkor feltételes kifejezést írhatunk be, amely – mikor igazzá válik – megszakítja a program futását. Ha 1-9 közé eső számot írunk be, akkor a nyomkövető a program, vagy eljárás nevét kérdezi meg, majd a sorszámot kell begépelnünk. (Ha olyan sorszámot adunk meg, amelyben nem szerepel parancs, az nem fog megszakításként működni.)

C (Clear) Törli a képernyőt

D (Display) A "SET" parancsok állapotát és az aktuális választást írja ki.
E (Expression) Kifejezések értékének ellenőrzésére alkalmas; érvényes kifejezés megadásakor annak aktuális értéke íródik ki.
G (Go) A program folyamatos végrehajtását kezdeményezi. A futás tart mindaddig, amíg egy megszakítási ponthoz nem ér, hiba nem jelentkezik, vagy Alt-D beütésével félbe nem szakítottuk.
L (List) Valamennyi PUBLIC változó neve és értéke megjelenik a képernyőn.
M (Memory variable) Valamennyi PRIVATE változó neve és értéke megjelenik a képernyőn a hozzájuk tartozó program- és eljárásnevekkel együtt.
P (Print) A nyomtató be/kikapcsolását teszi lehetővé.
Q (Quit) A program végrehajtását megszakítja, és visszatér az operációs rendszerhez.
R (Run) Lefuttat egy DOS programot.
S (Step) Minden "S" parancs hatására egyetlen programsor hajtódik végre. (Lépésenkénti üzemmód)
T (Trace) Az éppen futó program, vagy eljárás nevét és a sorok számát írja ki.
V (Variable) Egy memóriaváltozó értéke ellenőrizhető, vagy módosítható a "V" begépelése után.

Ha a program végrehajtása közben futás közbeni hiba lép fel, akkor a nyomkövető parancssora felcserélődik a hibaüzenet-sorral. (Run-time hibák leírását ld. 14.6.3 fejezetben.) Ha a hibaüzenetre "I"-vel válaszolunk, a hibakereső kapja vissza a vezérlést, így a hiba megfelelő korrigálása után a program végrehajtása folytatható lesz.

14.4.5 Az inkompatibilitások lehetséges feloldásai

14.4.5.1 A Clipper PUBLIC változó használata

Lehetőség van rá, hogy programjainkban definiáljunk egy Clipper nevű PUBLIC változót, és ennek értékétől függően illesszünk be programrészeket.

Ez egy logikai változó lesz, amely a "PUBLIC" deklaráláskor .T. értéket kap a Clipperben, a dBASE III PLUS-ban viszont .F. értéket. (Minden más PUBLIC változó a deklaráláskor .F. lesz mindkét nyelvben.)

Így ha ezt a változót "IF...ENDIF" parancsba foglaljuk, lehetőség van a dBASE III PLUS programok módosítására úgy, hogy azok mindkét módon futtathatók legyenek. (Figyeljünk rá, hogy ha a Clipper olyan dBASE III PLUS parancsot talál, amelyet nem támogat, akkor fordítási hibát jelez. Ilyenkor kommentezzük ki a kérdéses sorokat.) Például alkalmazhatjuk a következő megoldást:

```
PUBLIC Clipper
IF Clipper
 @ 10,10 GET a VALID a>10
ELSE
 DO WHILE .T.
  @ 10,10 GET a
```

```
  IF a<10
    EXIT
  ENDIF
 ENDDO
ENDIF
```

14.4.5.2 A Clipper Utility programjainak alkalmazása

Ezek a programok arra szolgálnak, hogy helyettesítsék a dBASE III PLUS interaktív parancsainak egy részét. Főleg az adatbázis, és formátum file-ok kezeléséhez nyújtanak szolgáltatásokat.

– A CREATE program

Ez a program új adatbázis file-ok létrehozását, vagy meglevő adatbázisok szerkezetének átalakítását segíti elő.

Használata:

C>CREATE [<d:>]<file-név>

ahol: "d:" a meghajtó azonosítója
"file-név" a létrehozni, vagy módosítani kívánt adatbázis neve a kiterjesztés (.DBF) nélkül.

A program ezután eligazít a használat felől. F1 begépelésével bármikor kérhetünk help-képernyőt, illetve Esc-pel megszakíthatjuk a program végrehajtását. Ne feledkezzünk meg róla, hogy egy létező adatbázis file szerkezetének módosításakor a rekordok megsemmisülnek. Ha ez szükséges, gondoskodjunk a file tartalmának elmentéséről a CREATE futtatása előtt.

– A REPORT program

Ez a program új jelentésszerkezeti file-ok létrehozásához, meglevők módosításához alkalmazható.

Használata:

C>REPORT [<d:>]<file-név>

ahol: "d:" a meghajtó azonosítója
"file-név" a létrehozni, vagy módosítani kívánt formátum file neve a kiterjesztés (.FRM) nélkül.

A program ezután eligazít a használat felöl. F1 és Esc a fentiekhez hasonlóan alkalmazható.

– A LABEL program

Ez a program új cimke-file létrehozását, vagy meglévő módosítását teszi lehetővé.

Használata:

C>LABEL [<d:>]<file-név>

ahol: "d:" a meghajtó azonosítója
"file-név" a létrehozni, vagy módosítani kívánt formátum file neve a kiterjesztés (.LBL) nélkül.

A program ezután eligazít a használat felől. F1 és Esc a fentiekhez hasonlóan alkalmazható.

– Az INDEX program

Ez a program az index file-ok létrehozására alkalmas.

Az INDEX program .PRG file formájában van jelen, ezért használat előtt le kell fordítani és szerkeszteni. (Azért van jelen forrásnyelven, hogy mintaprogramnak használhassuk néhány Clipper-ben alkalmazott technika megismeréséhez.)

Használata:
C>INDEX [<d:>]<adatbázisfile-név>
ahol: "d:" a meghajtó azonosítója
"adatbázisfile-név" az indexelni kívánt adatbázis neve a kiterjesztés (.DBF) nélkül.

A program kiírja az adatbázis mezőit, és megkérdezi az index file nevét, valamint a kulcsnak használt mező nevét.

– A LINE program

Ez a program kilistázza a programot a sorokat sorszámokkal ellátva.

Használata:
C>LINE <programnév.prg>[n]
ahol: "programnév" a kiíratni kívánt program neve a kiterjesztéssel együtt (.PRG)!
"n" opcionális paraméter, melynek hatására az "n" sorszámú sor íródik ki (a megelőző öt és azt követő tíz sorral együtt).
Alkalmazható a DOS output átirányító funkciója is.
Például listázáshoz nyomtatóra:
C>LINE Fopr.prg > PRN

14.4.5.3 A Clipper bővített könyvtárának használata

Létezik a felhasználói funkcióknak egy olyan gyűjteménye, amely – többek között – a legmesszebbmenő kompatibilitást biztosítja a dBASE III PLUS és a Clipper között. Ezen függvények szerzője Tom Retting. Valamennyi dBASE III PLUS funkciót tartalmazza, így ezek a Clipperből változtatás nélkül alkalmazhatók. Bővebb információk érdekében – és a számos egyéb szolgáltatás miatt is – tanulmányozzuk a Tom Retting's Library című dokumentációt.

14.5 Parancsok és funkciók öszefoglalása

Itt csak felsoroljuk a Clipperben használható parancsokat és funkciókat, a feladatuk részletezésétől eltekintünk.

Parancsok:

14.5.1 ? <kif. lista>
Egy vagy több kifejezés értékét írja a képernyő következő sorába.

14.5.2 ?? <kif. lista>
Egy vagy több kifejezés értékét írja a képernyő aktuális sorába.

14.5.3 @ <sor1>,<oszlop1>,<sor2>,<oszlop2> BOX <kar.kif>
Meghatározott karakterek használatával négyszöget rajzol a képernyő megadott helyére.

14.5.4 @ <sor1>,<oszlop1> CLEAR [TO <sor2>,<oszlop2>]
Törli a képernyő meghatározott részét.

14.5.5 @ <sor>,<oszlop> PROMPT <kar.kif> [MESSAGE <kar.kif>]
Menük készítéséhez alkalmazható.

14.5.6 @ <sor>,<oszlop> [SAY <kif> [PICTURE <kar.kif>]]
[GET <mem.vált> [PICTURE <kar.kif>] [RANGE <num.kif1>,<num.kif2>]
[VALID <log.kif>]]
Meghatározott sorban és oszlopban kiír vagy beolvas adatokat.

14.5.7 ACCEPT [<sztring>] TO <mem.vált>
Egy billentyűzetről beolvasott karakterláncot egy memóriaváltozóba tárol.

14.5.8 APPEND BLANK
Egy üres rekordot illeszt az adatbázis végére.

14.5.9 APPEND [<érv.kör>] [FIELDS <mezőlista>] FROM <file-név>
[FOR <feltétel>] [WHILE <feltétel>] [SDF/DELIMITED
[WITH BLANK/<határoló>/(<kar.kif>)]]
Az aktuális adatbázishoz új rekordot ad egy másik adatbázisból, vagy egy ASCII szöveg-file-ból.

14.5.10 AVERAGE [<érv.kör>] <mezőlista> TO <mem.vált lista>
[FOR <feltétel>] [WHILE <feltétel>]
Az aktuális adatbázis rekordjain egy vagy több kifejezés átlagát számítja, és memóriaváltozókba tárolja.

14.5.11 CALL <elj.név> [WITH <kif. lista>]
Lehetővé teszi külön fordított eljárások meghívását, szükség esetén paraméterek átadásával.

14.5.12 CANCEL
Megszakítja a program végrehajtását, lezárja a nyitott file-okat, és visszatér az operációs rendszerhez.

14.5.13 CLEAR
Letörli a képernyőt, törli a függőben maradt "GET" parancsokat és a kurzort a képernyő bal felső sarkába helyezi.

14.5.14 CLEAR ALL
Lezárja az összes adatbázis-file-t a kapcsolódó index-, formátum-, és memo-file-okkal együtt, és törli a memóriaváltozókat.

14.5.15 CLEAR GETS
Megszakítja a függőben maradt "GET" parancsokat.

14.5.16 CLEAR MEMORY
Törli az összes memóriaváltozót.

14.5.17 CLOSE ALL/ALTERNATE/DATA BASES/FORMAT/INDEX
Lezárja a kijelölt típusú file-t.

14.5.18 CONTINUE
Folytatja a függőben maradt "LOCATE" parancsot az aktuális file-ban.

14.5.19 COPY [<d:>]<file-azonosító1> TO [<d:>]<file-azonosító2>
Megkettőz bármilyen típusú file-t.

14.5.20 COPY STRUCTURE TO [<d:>]<file-név>
Létrehoz egy üres adatbázis-file-t az aktuális adatbázis struktúrájával.

14.5.21 COPY TO [<d:>]<file-név> [<érv.kör>][FIELDS <mezőlista>]
[FOR<feltétel>][WHILE <feltétel>][SDF/DELIMITED
[WITH BLANK/<határoló>/(<kar.kif>)]]
Az aktuális adatbázis egy részét, vagy egészét lemásolja egy másik file-ba.

14.5.22 COPY TO [<d:>]<file-név> STRUCTURE EXTENDED
Az aktuális adatbázisban meghatározott file-szerkezetet használja fel az új adatbázis létrehozásához.

14.5.23 COUNT [<érv.kör>][FOR <feltétel>][WHILE <feltétel>]
TO <mem.vált>
Összeszámolja az aktuális adatbázis rekordjait a feltételek szem előtt tartásával.

14.5.24 CREATE [<d:>]<file-név>
Egy bővített szerkezetű üres file-t hoz létre.

14.5.25 CREATE [<d:>]<file-név1> FROM [<d:>]<file-név2>
Egy bővített szerkezetű file alapján új adatbázist hoz létre.

14.5.26 DECLARE <tömb1>[<num.kif1>],<tömb2>[<num.kif2>]...
Egy vagy több memóriaváltozó-tömböt hoz létre.

14.5.27 DELETE [<érv.kör>][FOR <feltétel>][WHILE <feltétel>]
Az aktuális adatbázisban kijelöli a rekordokat törlésre.

14.5.28 DIR [<d:>][<út>][<skeleton>]
Kiírja a megadott meghajtón és/vagy tartalomjegyzékben levő file-ok nevét.

14.5.29 DISPLAY [OFF][<érv.kör>]<mezőlista>[FOR <feltétel>]
[WHILE <feltétel>]
Az aktuális adatbázis feltételeknek megfelelő rekordjaiból kiírja a kifejezések értékét.

14.5.30 DO <elj.név> [WITH <param. lista>]
Végrehajt egy eljárást, vagy programot szükség esetén paraméterek átadásával.

14.5.31 DO CASE...CASE...[OTHERWISE]...ENDCASE
Az első igaz értékü feltételnek megfelelő úton folytatja a program végrehajtását.

14.5.32 DO WHILE <feltétel> ...[EXIT]...[LOOP]...ENDDO
Ciklikus végrehajtását eredményezi, amíg a feltétel igaz értékű.

14.5.33 EJECT
A nyomtató írófejét a következő oldal elejére továbbítja, és a "PROW()" és "PCOL()" értékeit 0-ra állítja.

14.5.34 ERASE <file-azonosító>
Törli a megadott file-t a lemezről.

14.5.35 EXTERNAL <elj.név lista>
Szimbólumok meghatározása a program szerkesztő számara úgy, hogy elérhesse az overlay-be helyezett és/vagy makro-kkal hívott eljárásokat.

14.5.36 FIND <sztring>
A rekordmutatót az első olyan rekordra helyezi, amelynek indexkulcsa megegyezik a karakterlánccal.

14.5.37 FOR <mem.vált> = <num.kif1> TO <num.kif2>
[STEP<num.kif3>]...[EXIT]...NEXT
Ciklust valósít meg meghatározott növekménnyel.

14.5.38 FUNCTION <elj.név>...RETURN <kif>
Clipper-ben írt felhasználói függvényt határoz meg.

14.5.39 GO/GOTO <num.kif>/BOTTOM/TOP
Az aktuális adatbázisban a rekordmutatót a meghatározott rekordra helyezi.

14.5.40 IF...[ELSE]...ENDIF
A parancsok feltételtől függő végrehajtását valósítja meg alternatív lehetőséggel.

14.5.41 INDEX ON <kif> TO [<d:>]<file-név>
Az aktuális adatbázishoz index file-t hoz létre.

14.5.42 INPUT [<sztring>] TO <mem.vált>
Egy kifejezés beléptetését okozza egy memóriaváltozóba.

14.5.43 JOIN WITH <alias>/<kar.kif1> TO <file-név>/<kar.kif2>
FOR <feltétel> [FIELDS <mezőlista>]
Két adatbázis kiválasztott rekordjait egyesíti egy harmadik adatbázis file-ba.

14.5.44 KEYBOARD <kar.kif>
Egy meghatározott karakterlánccal tölti fel a billentyűzet pufferét.

14.5.45 LABEL FORM [<d:>]<file-név1>[<érv.kör>][FOR <feltétel>]
[WHILE <feltétel>][SAMPLE][TO FILE <file-név2>]
Egy .LBL file definíciói alapján címkéket nyomtat.

14.5.46 LIST [OFF][<érv.kör>]<mezőlista>[FOR <feltétel>]
[WHILE <feltétel>][TO PRINT/TO FILE<file-név>]
Az adatbázis rekordjairól írathatunk ki adatokat.

14.5.47 LOCATE [<érv.kör>][FOR <feltétel>][WHILE <feltétel>]
A rekordmutatót az első, a megadott feltételeknek megfelelő rekordra helyezi.

14.5.48 MENU TO <mem.vált>
A definiált "PROMPT" felhasználásával menüből való választást tesz lehetővé.

14.5.49 NOTE/*/&& <szöveg>
A programba nem végrehajtható szövegek illeszthetők be.

14.5.50 PACK
Az aktuális adatbázisból tartósan eltávolítja a törlésre megjelölt rekordokat.

14.5.51 PARAMETERS <mem.vált lista>
Memóriaváltozókat jelöl ki a továbbított mennyiségek, vagy referenciák fogadására.

14.5.52 PRIVATE <mem.vált lista>
A megjelölt változókat elrejti a magasabb szintü program, vagy hívó program elől.

14.5.53 PROCEDURE <elj.név>
Az eljárások kezdetét jelöli.

14.5.54 PUBLIC <mem.vált lista>/Clipper
A memóriaváltozókat általánosan elérhetővé teszi.

14.5.55 QUIT
Megszakítja a program végrehajtását, lezárja a nyitott file-okat, és visszatér az operációs rendszerhez.

14.5.56 READ
A "GET" parancsokkal együtt lehetővé teszi adatok bevitelét a képernyőről.

14.5.57 RECALL [<érv.kör>][FOR <feltétel>][WHILE <feltétel>]
Visszahívja a törlésre kijelölt rekordokat.

14.5.58 REINDEX
Újra felépíti a nyitott index file-okat.

14.5.59 RELEASE <mem.vált lista>/ALL [LIKE/EXCEPT <skeleton>]]
Törli a memóriaváltozókat.

14.5.60 RENAME <file-név1>.<ext> TO <file-név2>.<ext>
Egy létező file-nak új nevet ad.

14.5.61 REPLACE [<érv.kör>][<alias>->]<mező1> WITH <kif1>
[,<mező2> WITH <kif2>, ...][FOR <feltétel>][WHILE
<feltétel>]
Az adatmezők tartalmát megadott értékkel cseréli fel.

14.5.62 REPORT FORM [<d:>]<file-név1>[<érv.kör>][FOR <feltétel>] [WHILE <feltétel>][TO PRINT][TO FILE [<d:>]<file-név2>] [NOEJECT]
.FRM file-ban tárolt adatok alapján táblázatos kiírást hajt végre.

14.5.63 RESTORE FROM [<d:>]<file-név>
Visszahívja a .MEM file-ba elmentett adatokat.

14.5.64 RESTORE SCREEN
A "SAVE SCREEN" paranccsal elmentett képernyőt újra kirajzolja.

14.5.65 RETURN [<kif>]
A programok és eljárások záró utasítása, visszatér a hívó programhoz, vagy az operációs rendszerhez.

14.5.66 RUN/! <program>/<kar.kif>
Végrehajt egy DOS programot.

14.5.67 SAVE SCREEN
Az aktuális képernyőt egy pufferbe írja.

14.5.68 SAVE TO [<d:>]<file-név>/(<kar.kif)> [ALL [LIKE/EXCEPT <skeleton>]]
A memóriaváltozókat elmenti egy .MEM file-ba.

14.5.69 SEEK <kif>
A rekordmutatót az első olyan rekordra helyezi, amelynek indexe megegyezik a megjelölt kifejezéssel.

14.5.70 SELECT <munkater.>/<alias>/(<num.kif>)
Az aktuálisról egy másik munkaterületre vált át.

14.5.71 SET ALTERNATE TO [<d:>]<file-név>
Egy file-t hoz létre az outputok elmentésére.

14.5.72 SET ALTERNATE on/OFF
Meghatározza, hogy az output elmentődjék-e a kijejölt file-ba.

14.5.73 SET BELL on/OFF
Meghatározza, hogy az adatbevitel során kapunk-e hangjelzést.

14.5.74 SET COLOR TO [<normál>[,<kiemelt>][,<keret>]]
Meghatározza a képernyő színeit.

14.5.75 SET CONFIRM on/OFF
Meghatározza, hogy szükséges-e a "GET" befejezéséhez az Enter-t beütni.

14.5.76 SET CONSOLE on/OFF
Meghatározza, hogy az outputok megjelennek-e a képernyőn.

14.5.77 SET DECIMALS TO <num.kif>
A matematikai műveletek eredményeinek kiírásánál meghatározza a kijelzett tizedesjegyek számát.

14.5.78 SET DEFAULT TO <d:> [<út>]
Kijelöli a meghajtót és a könyvtárat a file-ok eléréséhez, létrehozásához és mentéséhez.

14.5.79 SET DELETED on/OFF
Meghatározza, hogy a törlésre kijelölt rekordokat kezeljük-e.

14.5.80 SET DELIMITERS on/OFF
Megszabja, hogy a "GET" mező határai láthatók legyenek-e.

14.5.81 SET DELIMITERS TO [<kar.kif>]/[DEFAULT]
Megszabja a "GET" mezők határoló karakterét.

14.5.82 SET DEVICE TO SCREEN/print
A "@" parancsok eredményeit képernyőre, vagy printerre küldi.

14.5.83 SET ESCAPE ON/off
Megszabja, hogy a program végrehajtása megszakítható-e Alt-C segítségével, illetve a "GET" megszakítható-e Esc billentyűvel.

14.5.84 SET EXACT on/OFF
Beállítja, hogy a karakterláncok összehasonlításánál pontos egyezés szükséges-e vagy nem.

14.5.85 SET FILTER TO [<feltétel>]
Az adatbázisból csak a feltételnek megfelelő rekordok kezelését engedélyezi.

14.5.86 SET FIXED on/OFF
Meghatározza, hogy függ-e a matematikai művelet minöségétől a kijelzett tizedesjegyek száma, vagy egyedül a "SET DECIMALS" parancs meghatározó.

14.5.87 SET FORMAT TO <elj.név>
Minden "READ" végrehajtásához aktivál egy formátum file-t.

14.5.88 SET FUNCTION < num.kif > TO < kar.kif >
A funkcióbillentyűk programozására alkalmas.

14.5.89 SET INDEX TO [< file-lista >]
Megnyitja a megadott index file-okat.

14.5.90 SET INTENSITY ON/off
A "GET" mezők színeit határozza meg.

14.5.91 SET KEY < num.kif > TO [< elj.név >]
Egy billentyűhöz kapcsolt eljárás végrehajtását teszi lehetővé bármely várakozó helyzetből.

14.5.92 SET MARGIN TO < num.kif >
Beállítja a baloldali margót a printeren és a képernyőn.

14.5.93 SET MESSAGE TO < num.kif >
Meghatározza, hogy a "PROMPT"-hoz kapcsolt üzenet a képernyő melyik sorában jelenik meg.

14.5.94 SET PATH TO [< útlista >]
Kijelöli a keresés útvonalát a Clipper számára a file-ok eléréséhez.

14.5.95 SET PRINT on/OFF
Meghatározza, hogy az output kiíródjon-e a printerre.

14.5.96 SET PROCEDURE TO [< d: >][< file-név >]
A fordítás során a megjelölt file-ból az összes eljárást az object file-hoz illeszti.

14.5.97 SET RELATION TO [< kulcs kif1 > / < RECNO() > / < num.kif1 >
INTO < alias1 >][,TO < kulcs kif2 > / < RECNO() > / < num.kif2 >
INTO < alias2 >]...
A kulcskifejezéseknek megfelelően összefűz két adatbázist.

14.5.98 SET UNIQUE on/OFF
Megszabja, hogy az index file-ba beírhatók-e azonos kulcsú rekordok.

14.5.99 SKIP [< num.kif1 > [ALIAS < munkater. > / < alias > /(< num.kif2 >)]]
A kiválasztott adatbázisban a rekordmutatót csökkenti, vagy növeli a megadott értékkel.

14.5.100 SORT [<érv.kör>] ON <mező1> [/A][/C][/D][<mező2>
[/A][/C][/D]]...TO [<d:>]<file-név> [FOR <feltétel>]
[WHILE <feltétel>]
Az aktuális adatbázis a feltételeknek megfelelő rekordjait sorbarendezve egy másik file-ba menti.

14.5.101 STORE <kif> TO <mem.vált lista>
Egy kifejezést egy vagy több memóriaváltozóba tárol.

14.5.102 SUM [<érv.kör>]<kif. lista> TO <mem.vált lista>
[FOR <feltétel>][WHILE <feltétel>]
Az aktuális adatbázisból kifejezések sorozatát összegzi és tárolja el memóriaváltozókba.

14.5.103 TEXT...ENDTEXT
Egy szövegblokkot ír ki a képernyőre.

14.5.104 TOTAL ON <kulcs kif> [<érv.kör>][FIELDS <mezőlista>
TO <file-név>] [FOR <feltétel>][WHILE <feltétel>]
Az aktuális adatbázisban összegzi a kiválasztott mezőket a megadott kulcsnak megfelelően, és átmásolja egy másik adatbázisba.

14.5.105 TYPE <file-név>
Egy szöveg-file tartalmát kiírja a képernyőre.

14.5.106 UPDATE ON <kulcs kif> FROM <alias> REPLACE <mező1>
WITH <kif1> [,<mező2> WITH <kif2>]...[RANDOM]
Az aktuális adatbázis-file-t felújítja egy másik adatbázisból.

14.5.107 USE [<file-név>][INDEX <file-lista>][ALIAS <alias>]
Megnyit egy létező adatbázis-file-t a hozzá tartozó index file-okkal együtt.

14.5.108 WAIT [<sztring>] [TO <mem.vált>]
Felfüggeszti a program végrehajtását egy billentyű beütéséig.

14.5.109 ZAP
Törli az összes rekordot az aktuális adatbázisból.

Funkciók:

14.5.110 ABS(<num.kif>)
A kifejezés abszolútértékét adja.

14.5.111 ASC(<kar.kif>)
A sztring első karakterének ASCII kódjával tér vissza.

14.5.112 AT(<kar.kif1>,<kar.kif2>)
Egy karakterlánc kezdő pozíciójával tér vissza egy másik karakterláncon belül.

14.5.113 BOF()
Egy file kezdetét mutatja.

14.5.114 CDOW(<dat.kif>)
Egy dátum-kifejezésből a hét napjának nevével tér vissza.

14.5.115 CHR(<num.kif>)
A megadott ASCII kódnak megfelelő karakterrel tér vissza.

14.5.116 CMONTH(<dat.kif>)
Egy dátum-kifejezésből a hónap nevével tér vissza.

14.5.117 COL()
A kurzor aktuális oszlop-pozíciójával tér vissza.

14.5.118 CTOD(<kar.kif>)
A karaktersztringként beléptetett, vagy tárolt dátumot dátum típusú változóvá alakítja.

14.5.119 DATE()
A rendszer-dátumot adja "mm/dd/yy" alakban.

14.5.120 DAY(<dat.kif>)
A dátum-változóból a hónap napjának numerikus értékét adja.

14.5.121 DELETED()
Egy logikai értékkel tér vissza, amely megmutatja, hogy az aktuális rekord ki van-e jelölve törlésre.

14.5.122 DOW(<dat.kif>)
A dátum-változóból egy numerikus értékkel tér vissza, amely megmutatja, hogy az a hét hányadik napja.

14.5.123 DTOC(<dat.kif>)
A dátum típusú változót karakteres változóvá alakítja.

14.5.124 DTOS(<dat.kif>)
A dátum-változóból "yyyymmdd" formátumú karakterláncot hoz létre.

14.5.125 EMPTY(<kif>)
.T.-val tér vissza, ha a változó üres.

14.5.126 EOF()
A file végét jelzi.

14.5.127 EXP(<num.kif>)
Az exponenciális függvény.

14.5.128 FIELDNAME(<num.kif>)
Megadja az aktuális adatbázis kiválasztott mezejének nevét.

14.5.129 FILE(<file-név>)
.T.-val tér vissza, ha a megadott file létezik.

14.5.130 FOUND()
Ha a legutóbbi "SEEK", "FIND", "LOCATE", vagy "CONTINUE" sikeres volt, akkor a visszatérő érték .T. lesz.

14.5.131 IF(<log.kif>,<kif1>,<kif2>)
A kifejezés feltételes végrehajtását segíti elő.

14.5.132 INKEY(<num.kif>)
A lenyomott billentyű ASCII kódjával tér vissza.

14.5.133 INT(<num.kif>)
Bármely numerikus értékből a tizedesjegyek elhagyásával egész számot állít elő.

14.5.134 LASTKEY()
Az utóljára lenyomott billentyű ASCII kódjával tér vissza.

14.5.135 LASTREC()
Az aktuális adatbázis rekordjainak számát adja.

14.5.136 LEN(<kar.kif>/<tömb>)
A sztringben, illetve tömbben levő karakterek, illetve tömbelemek számával tér vissza.

14.5.137 LOG(<num.kif>)
A természtes alapú logaritmus függvény.

14.5.138 LOWER(<kar.kif>)
A nagybetűs karaktereket kisbetűssé alakítja a sztringben.

14.5.139 MEMOEDIT(<kar.kif> [,<sor1>,<oszlop1>,<sor2>,<oszlop2>]
[,<log.kif>])
Kiírja, és szerkeszthetővé teszi a memóriamezőket.

14.5.140 MONTH(<dat.kif>)
A dátum típusú változóból a hónap nevével tér vissza.

14.5.141 PCOL()
A printer aktuális oszlop-pozícióját adja.

14.5.142 PROCLINE()
A futó program forráskódjának aktuális sorszámával tér vissza.

14.5.143 PROCNAME()
A futó program, vagy eljárás nevével tér vissza.

14.5.144 PROW()
A printer aktuális sor-pozícióját adja.

14.5.145 RECNO()
A kiválasztott adatbázis aktuális rekordjának sorszámával tér vissza.

14.5.146 REPLICATE(<kar.kif>,<num.kif>)
A karakterlánc meghatározott számú ismétlését végzi.

14.5.147 ROUND(<num.kif1>,<num.kif2>)
A számokat megadott számú tizedesjegyre kerekíti.

14.5.148 ROW()
A kurzor aktuális sor-pozíciójával tér vissza.

14.5.149 SECONDS()
A rendszeridő "másodpercek.századok" alakú értékével tér vissza.

14.5.150 SELECT()
Az aktuális adatbázis számával tér vissza.

14.5.151 SPACE(<num.kif>)
Előállít egy meghatározott számú szóközökből álló sztringet.

14.5.152 SQRT(<num.kif>)
A megadott pozitív szám négyzetgyökét adja.

14.5.153 STR(<num.kif1> [<num.kif2>,<num.kif3>])
A numerikus kifejezést karakterlánccá alakítja.

14.5.154 SUBSTR(<kar.kif>,<num.kif1>[,<num.kif2>])
A karakterlánc meghatározott részét állítja elő.

14.5.155 TIME()
A rendszeridővel tér vissza "hh:mm:ss" alakban.

14.5.156 TRANSFORM(<kif>,<kar.kif>)
Egy karakterlánc meghatározott alakját állítja elő.

14.5.157 TRIM(<kar.kif>)
Eltávolítja a szóközöket a sztring végéről.

14.5.158 TYPE("<kar.kif>")
Egy memóriaváltozó, kifejezés, vagy mező adattípusával tér vissza.

14.5.159 UPDATED()
Ha az utolsó "READ" megváltoztatta a "GET" adatait, akkor a visszatérő érték .T. lesz.

14.5.160 UPPER(<kar.kif>)
A kisbetűs karaktereket nagybetűssé alakítja.

14.5.161 VAL(<kar.kif>)
Egy karakterláncot számmá alakít.

14.5.162 WORD()
A "CALL" parancs numerikus argumentumait egész típusúvá alakítja.

14.5.163 YEAR(<dat.kif>)
A dátum-változóból az év teljes (4 karakteres) értékével tér vissza.

14.6 Hibaüzenetek

14.6.1 Fordítás

A Clipper fordító hibaüzenetei

ACCEPT/INPUT error
APPEND error
ASSIGNMENT error
AVERAGE error
AVERAGE number of expressions and variables
CALL error
CLEAR error
CLOSE error

COPY error
COUNT error
CREATE error
DELETE error
DO error
FILTER error
FOR error
LABEL error
LIST error
LOCATE error
PARAMETER error
RECALL error
RELATION error
RELEASE error
REPLACE error
RESTORE error
SAVE error
SET not recognized
SET switch error
SUM error
USE error
WAIT error

A következő üzenetek speciális hibákhoz, illetve helyzetekhez kapcsolódnak:

case w/o do case
Hiányzik a "DO CASE" utasítás

else w/o if
Hiányzik az "IF" utasítás

endcase w/o do case
Hiányzik a "DO CASE" utasítás

enddo w/o while
Hiányzik a "DO WHILE" utasítás

endif w/o if
Hiányzik az "IF" utasítás

exit w/o do while
Hiányzik a "DO WHILE" utasítás

fatal at < n > - invalid procedure mode
Két eljárásnak ugyanaz a tartalma, vagy a neve. Ez a hibaüzenet csak a képernyőn jelenik meg, nem irányítható át file-ba.

illegal device
A "SET DEVICE TO < SCREEN/PRINT > " utasításban szabálytalan opciót talált.

illegal value
Az értékadó utasítás baloldalán szabálytalan név van.

loop w/o do while
A "LOOP" nem "DO WHILE...ENDDO" szerkezetben van.

missing 2nd quote
A befejező szöveghatároló hiányzik.

next w/o for
Hiányzik a "FOR" utasítás.

otherwise w/o do case
Hiányzik a "DO CASE" utasítás.

rest of line ignored
A megadott sor maradék részének szintaxisát a fordító nem támogatja, figyelmen kívül hagyja.

symbol redefinition error
Ez a hibaüzenet általában akkor jelenik meg, ha az eljárás file és a benne lévő eljárás neve megegyezik.

too many constants
Túl sok a konstans a programban. Fordítsuk részletekben a programot, majd fűzzük össze a részeket.

too many symbols
Túl sok a szimbólum a programban. Fordítsuk részletekben a programot, majd fűzzük össze a részeket.

unbalanced DO CASE
Hiányzik az "ENDCASE".

unbalanced DO WHILE
Hiányzik az "ENDDO".

unbalanced FOR/NEXT
Hiányzik a "NEXT".

unbalanced IF/ELSE
Hiányzik az "ENDIF".

verb not recognized
A parancsot a Clipper nem támogatja.

14.6.2 Szerkesztés

Figyelmeztető kódok

1 Kísérletet tettünk egy 16 bit-es cím használatára olyan memóriahelyre hivatkozva, mely 64 kByte-nál távolabb helyezkedik el, vagy megkíséreltünk egy 256 kByte-nál távolabbi címre hivatkozni egy 8 bites címmel. A másik lehetőség az, hogy a hivatkozott cím lejjebb van a memóriában, mint a feltételezett szegmens regiszter értéke. Például: közeli hívás az aktuális CS-nél lejjebb lévő memóriacímre, vagy rossz szegmens címet használó távoli hívás. A ténylegesen alkalmazott távolság nem ebbe a tartományba fog esni, így a program valószínűleg nem fogja helyesen megcímezni a célt.

2 A DOS alatt a program kívánt veremcíme a .EXE file fejlécében van. Ezt a program végrehajtásakor az SS és SP regiszterek beállítására használjuk. A PLINK86 olyan verem szegmenst keres, mely a fordító vagy assembler használatát mutatja. Ha talál ilyet, akkor annak végleges címe a fejlécbe kerül. Ha nem, akkor oda nullák íródnak, és a program valószínűleg nem fog helyesen működni, hacsak nem maga állítja be az SS és SP regisztereket. A program még ilyenkor is elromolhat, ha az érvényes verem beállítása előtt megszakítást kap.

3 A GROUP szegmensek olyan csoportja, amelynek 64 kByte memóriaterületen belül kell elhelyezkednie. Ez 16 bites címek használatát teszi lehetővé a csoporton belüli elemek elérésére. A csoport egy része azonban nem érhető el, ha a csoport túl nagy, ilyenkor csökkenteni kell a csoport méretét. Néha megjelenhet ez a figyelmeztetés akkor is, ha biztosak vagyunk abban, hogy a csoportot alkotó szegmensek együttes mérete kisebb 64 kByte-nál. Nézzük meg a memóriatérképet, hogy a csoportban nincsenek-e a szegmenseket elválasztó közbenső szegmensek. Használjunk overlay-ket a kód-csoportok memóriaigényének csökkentésére.

4 A megnevezett modult a "MODULE" parancsban használtuk, azonban a PLINK86 nem talált az input file-ok között ilyen nevű modult.

5 A 8086-os mikroprocesszor kettős címzést alkalmaz. Egy hosszú cím (FAR) távolság része az első rész által kiválasztott 64 kByte nagyságú szegmensre vonatkozik. Noha a memóriacímtől való relativ távolság a szerkesztés alkalmával meghatározható, a kezdő cím abszolút cím lesz a memóriában, és azt annak megfelelően kell beállitani, hogy az operációs rendszer hová tölti a programot a memóriába. Ezt a DOS végzi el, mikor a programokat a végrehajtás előtt betölti. A PLINK86 outputja egy hosszú cím, mely a program kezdőcímére vonatkozik.

6 Ez a figyelmeztetés nem fordulhat elő a DOS használatakor.

7 A megnevezett modul a PLINK86 számára ismeretlen típusú rekordot tartalmaz. A PLINK86 az egész rekordot átugorja. Az üzenetek kiírása megszűnik, miután néhányat kinyomtattunk. Ez általában akkor fordul elő, ha a .PRG kiter-

jesztést is megadtuk, amikor a DOS batch file-ba (CL.BAT) beírtuk a program nevét.

8 Egy object file-ban minden rekord a rekord végén tartalmaz egy ellenőrző mezőt az érvényesség felülvizsgálata céljából. Ez az üzenet jelzi, hogy az ellenőrző összeg hibás volt, de a szerkesztés folytatódik.

9 Egy tárgy file-ban minden rekordot megelőz egy szó, mely megadja a rekord méretét (ellenőrizhető a DUMP utility segítségével is). Ez a hiba azt jelenti, hogy a PLINK86 elérte a rekord végét, és úgy találta, hogy a feldolgozott Byte-ok száma különbözik a megadott mérettől. A tárgykódú file valószínűleg tönkremehet, de a PLINK86 megkísérli a szerkesztést.

10 Egy 16 bites címet használtunk, noha a célprogram 64 kByte-nál nagyobb távolságra van a keretprogramtól (vagy 8 bites a címet 256 kByte-nál nagyobb távolság esetén). Így a célprogramot nem értük el a kívánságunk szerint. A ténylegesen használt címnek meg kell felelnie a szükséges távolságnak. Ha van 64 kByte-nál nagyobb csoport, akkor csökkenteni kell a méretét a hozzáférés érdekében.

11 A megnevezett szimbólum kétszer van meghatározva, holott az összefűzendő programban csak egy meghatározást adhatunk minden egyes "PUBLIC" szimbólumra. A PLINK86 figyelmen kívűl hagyja a kettős meghatározást, megtartja az elsőt, és folytatja a szerkesztést.

12 A megnevezett "PUBLIC" szegmens több csoporthoz van hozzárendelve. Egyik csoport a szegmenst az egyik csoportban helyezte el, majd később egy másik modul egy másik csoportot ad meg ugyanarra a szegmensre. A szegmensekre történő hivatkozási címek nem lesznek helyesek.

13 A megnevezett szegmenst először "PUBLIC" szegmensként határoztuk meg, majd később, mint közönséges blokkot (vagy fordítva). Győződjünk meg arról, hogy egy szegmens valamennyi meghatározását ugyanolyan típusúra változtassuk.

14 A megnevezett modulon belül kettős verem szegmenst határoztunk meg. A PLINK86 csak az utolsó verem-meghatározást használja az .EXE fejlécben a verem megadására. Ha ez a verem szegmens üres vagy túl kicsi, a program valószínűleg megáll a végrehajtás során. Minthogy a PLINK86 az előző verem-meghatározást figyelmen kívűl hagyja, nincs értelme egynél több verem szegmens használatának. Győződjünk meg arról, hogy valamennyi verem szegmens meghatározás ugyanazt a nevet és csoportnevet használja, s azután ezeket egyetlen szegmensben egyesítjük, melynek mérete az eredeti szegmensek méretének összege lesz.

Hibaüzenetek

1 Túl sok "@" file van egymásba ágyazva (csak három szint mgengedett). Ellenőrizzük, nincs-e ciklus (önmagára hivatkozás) a "@" file hivatkozásokban.

2 A "@" file-ok olvasásakor lemezhibát talált a program. Próbáljuk újraépíteni a file-t.

3 A PLINK86 nem találja a "@" után megnevezett file-t.

5 Az inputként megadott tétel túl hosszú. A maximálisan megengedett hosszúság 64 karakter.

6 Érvénytelen számjegy egy számban.

10 Érvénytelen file-név. Az inputnak a DOS-ban érvényes file-nevet kell tartalmaznia.

11 Várakozás egy utasításra. Utasítást megkezdő kulcs-szónak kell itt állnia.

14 Várakozás egy azonosítóra. Ezen a ponton egy rész, modul, szegmens vagy szimbólum nevét kell beléptetni.

15 Várakozás egyenlőségjelre.

16 Várakozás egy értékre. Ezen a ponton egy 16 bites mennyiségnek kell megjelennie.

17 Nem adtunk meg file-okat az összefűzéshez. A "FILE" utasítást kell használnunk, és legalább egy input file-t meg kell adnunk.

18 Várakozás egy befejező zárójelre a "CLASS" utasítás után. A szegmens nevek felsorolását zárójelek közé kell zárni.

30 Nem hozható létre munkafile. Valószínűleg nincs hely a lemez katalógusában.

31 A munkafile írása közben I/O hiba fordult elő.

32 A munkafile olvasása közben I/O hiba fordult elő.

33 A munkafile áthelyezése közben I/O hiba fordult elő.

34 Túl sok tárgyfile-t (szimbólumot, szegmenst, csoportot, stb.) határoztunk meg egy programban (hozzávetőleg 35000 hozható létre). A program túl nagy a szerkesztéshez.

41 A jelzett object file végét váratlanul értük el. A file-ból valószínűleg levágtunk egy darabot, amikor olyan programmal másoltuk, mely feltételezi, hogy a Ctrl-Z a file vége. Ez általában akkor fordul elő, amikor egy DOS batch file-ban (CL.BAT) a programnevet kiterjesztéssel adtuk meg.

42 Hiba az input tárgykódú file olvasásakor.

43 A PLINK86 nem találja a megnevezett tárgykódú (object) file-t.

45 Nem hozható létre output lemezfile. A lemez katalógusa valószínűleg tele van, vagy a lemez írásvédett.

46 Érvénytelen output file típus. Ha megadtuk a kiterjesztést, akkor az .EXE vagy .CMD kell hogy legyen.

47 Lemezírási hiba az output file-ban. A lemez valószínűleg tele van vagy írásvédett, vagy valamilyen hardware hiba történt.

48 Lemezolvasási hiba az output file-ban. A lemez valószínűleg tele van vagy írásvédett, vagy valamilyen hardware hiba történt.

49 Az output file nem zárható le. A lemez valószínűleg tele van vagy írásvédett, vagy valamilyen hardware hiba történt.

50 Memóriatérkép file nem hozható létre. A lemez valószínűleg tele van vagy írásvédett, vagy valamilyen hardware hiba történt.

51 Határozatlan szimbólum. Hibásan gépeltük az eljárás, vagy funkció nevét. A felsorolt szimbólumok több modulban is szerepelhetnek, de azokat egyikben sem definiáltuk.

54 Nincs elegendő memória a PLINK86 futtatásához. A PLINK86 memóriaigénye legalább 256 kByte.

57 Probléma van az OVERLAY.LIB file-lal. Győződjünk meg arról, hogy a legújabb változatot használjuk. Ha a saját OVERLAY.LIB file-unkat használtuk, akkor az helytelenül formattált.

58 A verem szegmens túl nagy (nagyobb 64 kByte-nál) Ne felejtsük el, hogy az egyes modulokban meghatározott verem szegmensek közös szegmensekké lesznek összekapcsolva.

61 Egy LTL szegmens jelent meg egy Intel modulban. A PLINK86 nem fogadja el ezeket inputként.

62 Egy REGINT rekord az .EXE file-formátum által nem támogatott módon írt elő egy regisztert. Kizárólag a CS:IP és az SS:IP adható meg előre a REGINT rekordban.

63 Egy LIDATA alrekord ismételt nulla összeggel rendelkezik.

64 Egy Intel formátumú könyvtári tárgykódú file-t használtunk helytelenül kialakított index-szel. Segíthet, ha a könyvtárat a PLINK86 használatával újraszerkesztjük.

65 Egy abszolút kezdőcímet írtunk elő egy adott modulban. A PLINK86 valamely szegmenshez viszonyított relatív cím megadását igényli.

66 Ez a modul a PLINK86 által nem kezelt típusú elemcsoportot alkalmaz. Jelenleg csak szegmensek foglalhatók csoportokba.

70 Érvénytelen memóriahelyet adtunk meg egy beállítási pontnak. Egy relatív szegmens-beállítási pont LOC mezeje nagyobb négynél.

71 Érvénytelen memóriahelyet adtunk meg egy beállítási pontnak. Egy saját szegmens-beállítási pont LOC mezeje nagyobb egynél.

73 A keret megadási típusát a PLINK86 nem támogatja (5 vagy 7).

80 Az overlay-ket túl mélyen ágyaztuk be. Túl sok szint van az overlay szerkezetben.

81 Túl sok "ENDAREA" utasítás van - számuk nagyobb a már lezárt "BEGINAREA" utasítások számánál.

82 A "BEGINAREA" és az "ENDAREA" utasítások száma nem egyforma.

83 Az összefűzött program nem fér be az 1 MByte-os címterületbe, ami az Intel 8086 mikroprocesszor család által támogatott maximum. Overlay-k használatára lesz szükség a program memóriaigényének csökkentésére.

200-226
Ezek a PLINK86-nak hibáit jelzik. Nem a felhasználók idézik elő ezeket a problémákat, és valószínűtlen, hogy meg tudnák oldani. Először próbáljuk ismételten lefuttatni a PLINK86-ot ha nem működik, másoljuk le ismét a PLINK86-ot a Clipper lemezről, és futtassuk le újra. Ha a hiba továbbra sem szűnik meg, jegyezzük fel a hibakódot és lépjünk kapcsolatba a Nantucket-tel.

14.6.3 Futás közben kiadott hibaüzenetek

Ha ezen hibák közül valamelyik a végrehajtás közben előfordul, a program futása megszakad, és a következő sor jelenik meg a képernyő tetején:

proc: line: (leírás) (Q/A/I)

ahol: "proc" annak a programnak a neve, amelyben a hibát tapasztaltuk.
"line" az utasítás sorának száma a programon belül
"(leírás)": a hibák típusa
"(Q/A/I)" a folytatáshoz ezen betűk valamelyikét kell beléptetnünk:
(Q)uit — file-ok lezárása, a végrehajtás leállítása és visszatérés a DOS-hoz.
(A)bort — a végrehajtás leállitása, és visszatérés a DOS-hoz.
(I)gnore — a hiba figyelmen kívül hagyása, és a végrehajtás folytatása.

0 - Rendszer hibák
Kapcsoljuk ki a számítógépet, és próbáljuk újra.

1 - DOS hibák
2 A file nem elérhető.
4 Túl sok a nyitott file.
8 Kevés memória.

2 - Adatbázis hibák
1 .DBF fejcímke olvasási hiba.
2 Nincs nyitott adatbázis.

3 - Matematikai hibák
1 Nullával való osztás.

4 - Típus-ütközések
Különböző adat típusokon kíséreltük meg műveletek elvégzését. Az ilyen fajta hibák esetében kiírt üzenet tartalmazza a hibát okozó operátor nevét.

5 - Definiálatlan
2 Definiálatlan alias-név
4 Definiálatlan mezőnév
5 Definiálatlan változónév

6 - Makro
1 alias
2 helytelen a függvény-argumentumok száma
3 érvénytelen alias-név
4 érvénytelen függvény-argumentum
5 érvénytelen mezőnév
6 érvénytelen összerendelés
7 hiányzik az operátor
8 operandus-hiba
9 páratlan számú idézőjel
10 túl sok literal
11 túl sok konstans
12 érvénytelen szimbólum
13 páratlan számú zárójel

14.7 A '86 őszi változat (Többfelhasználós környezet)

A lokális hálózatok (LAN) lehetővé teszik osztott adatbázisok és perifériák ellenőrzött használatát több felhasználó számára.

Az adatok megosztott hozzáférését a hálózati operációs rendszer biztosítja azáltal, hogy több felhasználónak engedélyezi a file megnyitását egyidőben. Az adatok használatának ellenőrzése pedig olyan funkciók alkalmazásának következménye, amelyek időlegesen felülbírálják az adatok egyidejü használhatóságát, és egyetlen felhasználó ellenőrzése alá vonják az adatbázisokat.

A Clipper '86 őszi változata a következő eszközöket biztosítja a hálózati környezetben futó programok fejlesztéséhez:

Parancsok:

14.7.1 SET EXCLUSIVE ON/off/(<log.kif>)
Meghatározza, hogy az ezután megnyitott file-ok használata osztott, vagy kizárólagos.

14.7.2 SET PRINTER TO [<device>/[<d:>]<file-név>/(<kar.kif>)]
Meghatározza a kinyomtatott output számára a célkészüléket. (Ez lehet egy hálózati nyomtató nyomtatási sora is.)

14.7.3 UNLOCK [ALL]

Felszabadítja a rekordokra, vagy file-okra előzőleg kiadott zárolásokat.

Funkciók:

14.7.4 FLOCK()

Megkísérli a kiválasztott adatbázis-file zárolását. A visszatérő érték egy logikai változó, amely .T. lesz ha a zárolás sikeres volt. Ekkor az adatbázis másik felhasználó számára csak olvasásra hozzáférhető. (A Clipper - a dBASE III PLUS-tól eltérően - nem zárolja automatikusan a "SET RELATION" paranccsal kapcsolt file-okat!)

14.7.5 NETERR()

Amennyiben a "USE", "USE EXCLUSIVE", vagy az "APPEND" utasítás sikertelen volt, .T. értékkel tér vissza.

14.7.6 NETNAME()

Egy karaktersztringgel tér vissza, amely az aktuális munkahely azonosítóját tartalmazza. Csak az IBM PC Network használatánál működik, egyébként üres sztringgel tér vissza.

14.7.8 RLOCK()/LOCK()

Megkísérli zárolni az aktuális rekordot a kiválasztott munkaterületen. Sikeres zárolás esetén .T. értéket ad. (A Clipper - a dBASE III PLUS-tól eltérően - nem zárolja automatikusan a "SET RELATION" paranccsal kapcsolt rekordokat!)

A Clipper akkor futtatható hálózati környezetben, ha a számítógép rendelkezik a PC DOS 3.1 vagy annál fejlettebb változatával. A DOS-file-hívások mindíg "EXCLUSIVE" használatot eredményeznek. A többfelhasználós programok fejlesztése előtt meg kell ismerkednünk a hálózat sajátosságaival, tisztázni kell a következő kérdéseket:

- Mikor kell zárolni a rekordokat és file-okat?
- Hogyan lehet osztott módban megnyitni a file-okat?
- Milyen a zárolás mechanizmusa?
- Hogyan kíséreljük meg a zárolást?
- Hogyan oldjuk fel a sikertelen zárolást?

Anélkül, hogy a kérdések kimerítő elemzésére vállalkoznánk, néhány alapvető ismeretet az alábbiakban felsorolunk:

1.) A parancsok végrehajtásánál a file-ok megnyitásának módja.

– A file-t OLVASSÁK, és OSZTOTT MODBAN nyitják meg:

APPEND FROM [<d:>]<file-név>
CREATE...FROM [<d:>]<file-név>
LABEL FORM [<d:>]<file-név>
REPORT FORM [<d:>]<file-név>
RESTORE FROM [<d:>]<file-név>

TYPE [<d:>]<file-név>
UPDATE...FROM [<d:>]<file-név>

– A file-ba IRNAK, és EXCLUSIVE MODBAN nyitják meg:
COPY STRUCTURE TO [<d:>]<file-név>
COPY TO [<d:>]<file-név>
CREATE [<d:>]<file-név>
INDEX ON...TO [<d:>]<file-név>
JOIN...TO [<d:>]<file-név>
SAVE TO [<d:>]<file-név>
SET ALTERNATE TO [<d:>]<file-név>
SORT...TO [<d:>]<file-név>
TOTAL...TO [<d:>]<file-név>

2.) A hálózati környezet hatása a parancsokra.

Parancs		Szükséglet
@...SAY...GET		RLOCK()
APPEND FROM		USE...EXCLUSIVE v. FLOCK()
DELETE	(egy rekord)	RLOCK()
DELETE	(több rekord)	USE...EXCLUSIVE v. FLOCK()
PACK		USE...EXCLUSIVE
RECALL	(egy rekord)	RLOCK()
RECALL	(több rekord)	USE...EXCLUSIVE v. FLOCK()
REINDEX		USE...EXCLUSIVE
REPLACE	(egy rekord)	RLOCK()
REPLACE	(több rekord)	USE...EXCLUSIVE v. FLOCK()
UPDATE ON		USE...EXCLUSIVE v. FLOCK()
ZAP		USE...EXCLUSIVE

A '86 őszi változat új utasításai és függvényei

14.7.9 SET CENTURY on/OFF/(<log.kif>)
Meghatározza, hogy a dátum típusú változók kiírásánál megjelenjen-e a század.

14.7.10 SET DATE AMERICAN/ANSI/BRITISH/ITALIAN/FRENCH/GERMAN
Megszabja a dátum formátumát.

14.7.11 SET ORDER TO (<num.kif>)
Beállítja, hogy az aktuális file hanyadik indexe legyen a vezérlő index.

14.7.12 SET SCOREBOARD on/OFF/(<log.kif>)
Megszabja, hogy a Clipper üzenetei megjelenjenek-e a legfelső sorban.

14.8 A '87 nyári változat

A Clipper '87 nyári változata a következő parancsokkal és funkciókkal bővíti eszköztárunkat:

Parancsok:

14.8.1 SET CURSOR on/OFF/(<log.kif>)
Be-, illetve kikapcsolja a kurzort.

14.8.2 SET SOFSEEK on/OFF/(<log.kif>)
Meghatározza, hogy a keresés relatív legyen-e. (Relatív a keresés, ha az adott kulcsú rekord hiánya esetén a következő, az adott kulcsnál nagyobb rekordra áll.)

14.8.3 SET WRAP on/OFF/(<log.kif>)
Amennyiben a kapcsoló állása "ON", és az utolsó menüpont az aktuális, a lefelé mutató nyíl megnyomására az első menüpont lesz az aktuális.

Függvények:

14.8.4 ACHOICE(<sor1>,<oszlop1>,<sor2>,<oszlop2>,<tömb1>
[,<tömb2>[,<kar.kif>[,<num.kif1>[,<num.kif2>]]]])
Leegyszerűsíti a menüszerkesztés folyamatát. A visszatérő érték a kiválasztott menüpont sorszáma.

14.8.5 ACOPY(<tömb1>,<tömb2>[,<num.kif1>[,<num.kif2>[,<num.kif3>]]])
Tömb meghatározott része másolható egy másik tömbbe. Nem ad vissza értéket.

14.8.6 ADEL(<tömb>,<num.kif>)
Kihagyja a tömb egy meghatározott elemét olymódon, hogy az utána következő elemek az eggyel kisebb sorszámúra másolódnak. Az utolsó elem értéke definiálatlan lesz. Nem ad vissza értéket.

14.8.7 ADIR(<tartalomjegyzék skeleton>[,<tömb1>[,<tömb2>
[,<tömb3>[,<tömb4>[,<tömb5>]]]]])
Az aktuális tartalomjegyzéket, és a file-ok jellemzőit tölti bele a tömbökbe. A visszatérő érték a file-ok száma.

14.8.8 AFIELDS([<tömb1>[,<tömb2>[,<tömb3>[,<tömb4>]]]])
Az aktuális adatbázis mezőinek jellemzőit tölti bele a tömbökbe. A visszatérő érték a mezők száma.

14.8.9 AFILL(<tömb>,<kif>[,<num.kif1>[,<num.kif2>]])
Feltölt egy tömböt megadott értékekkel. Nem ad vissza értéket.

14.8.10 AINS(<tömb>,<num.kif>)
Beszúr egy definiálatlan elemet a tömbbe úgy, hogy az utána következő elemek az eggyel nagyobb sorszámúra másolódnak. Nem ad vissza értéket.

14.8.11 ALIAS([<num.kif>])
A meghatározott adatbázis alias-nevével tér vissza.

14.8.12 ASCAN(<tömb>,<kif>[,<num.kif1>[,<num.kif2>]])
Egy meghatározott értéket keres a tömbben. Annak ez elemnek a sorszámával tér vissza, amelyben a keresett érték előfordult.

14.8.13 ASORT(<tömb>[,<num.kif1>[,<num.kif2>]])
Növekvő sorrendbe rendezi a tömb elemeit. Nem ad vissza értéket.

14.8.14 DBEDIT([<sor1>[,<oszlop1>[,<sor2>[,<oszlop2>]]]]
[,<tömb1>][,<kar.kif>][,<tömb2>][,<tömb3>][,<tömb4>]
[,<tömb5>][,<tömb6>][,<tömb7>])
Kiírja, és szerkeszthetővé teszi egy vagy több adatbázis rekordjait.

14.8.15 FCLOSE(<num.kif>)
Lezár egy DOS file-t. Sikeres lezárás esetén .T. értékkel tér vissza.

14.8.16 FCOUNT()
A kiválasztott adatbázis mezőinek számával tér vissza.

14.8.17 FCREATE(<kar.kif>[,<num.kif>])
Létrehoz egy 0 hosszúságú DOS file-t. Az új file a DOS file-kezelő számával tér vissza.

14.8.18 FERROR()
A DOS file-műveletek befejezése után visszaadja az előfordult hiba kódját.

14.8.19 FOPEN(<kar.kif>[,<num.kif>])
Megnyit egy DOS file-t. A file DOS file-kezelő számával tér vissza.

14.8.20 FREAD(<num.kif1>,@<mem.vált>,<num.kif2>)
Karaktereket olvas a file-ból egy karakteres memóriaváltozóba. A visszatérő érték a sikeresen beolvasott karakterek száma.

14.8.21 FREADSTR(<num.kif1>,<num.kif2>)
Karaktereket olvas a file-ból addig, amíg ASCII 0-t nem talál. A visszatérő érték a beolvasott karaktersztring.

14.8.22 FSEEK(<num.kif1>,<num.kif2>[,<num.kif3>])
Beállítja a file-mutatót a megadott pozícióra. A mutató új értékével tér vissza.

14.8.23 FWRITE(<num.kif1>,<mem.vált>[,<num.kif2>])
Kiírja a memóriaváltozó tartalmát a megadott file-ba. A kiírt karakterek számával tér vissza.

14.8.24 HARDCR(<kar.kif>)
A CHR(141) értéket CHR(13) -ra cseréli a memóriamezőkben. A visszatérő érték az átalakított karaktersztring.

14.8.25 INDEXEXT()
Megadja az aktuális indexfile-ok típusát. (.NDX/.NTX)

14.8.26 INDEXKEY(<num.kif>)
A kiválasztott index kucskifejezésével tér vissza.

14.8.27 INDEXORD()
A "SET ORDER" paranccsal megadott vezérlő index számával tér vissza.

14.8.28 ISALPHA(<kar.kif>)
.T. értékkel tér vissza, ha a karaktersztring első karaktere alfabetikus.

14.8.29 ISCOLOR()
.T. értékkel tér vissza, ha színes grafikus kártya (CGA) van a számítógépben.

14.8.30 LTRIM(<kar.kif>)
A megadott karaktersztringről levágja a kezdő szóközöket.

14.8.31 MAX(<num.kif1>/<dat.kif1>,<num.kif2>/<dat.kif2>)
A megadott paraméterek közül a nagyobbikkal tér vissza.

14.8.32 MEMOREAD(<kar.kif>)
Beolvassa egy lemezes szöveg-file tartalmát.

14.8.33 MEMORY(<num.kif>)
A szabad pool-terület nagyságát adja.

14.8.34 MEMOTRAN(<kar.kif1>[,<kar.kif2>[,<kar.kif3>]])
Cseréli a CR/LF párokat.

14.8.35 MEMOWRIT(<kar.kif1>,<kar.kif2>)
Kiírja a karaktersztringet egy meghatározott lemezes file-ba. .T. értékkel tér vissza, ha a művelet sikeres volt.

14.8.36 MIN(<num.kif1>/<dat.kif1>,<num.kif2>/<dat.kif2>)
A megadott paraméterek közül a kisebbikkel tér vissza.

14.8.37 MLCOUNT(<kar.kif>,<num.kif>)
Megadja egy karaktersztringben vagy memo-mezőben a sorok számát.

14.8.38 NEXTKEY()
A billentyűzet-puffer következő karakterének értékével tér vissza, anélkül hogy azt eltávolítaná.

14.8.39 PCOUNT()
Egy függvénybe, vagy eljárásba átadott paraméterek számával tér vissza.

14.8.40 RAT(<kar.kif1>,<kar.kif2>)
Egy megadott karaktersztringben keres egy másik karaktersorozatot úgy, hogy jobbról balra halad. A visszatérő érték, a kezdő pozíció.

14.8.41 READVAR()
Megadja az aktuális "GET"/"MENU" változó nevét.

14.8.42 SCROLL(<num.kif1>,<num.kif2>,<num.kif3>,<num.kif4>,
<num.kif5>)
A képernyő megadott részén belül scroll-ol. Nem ad vissza értéket.

14.8.43 SETPRC(<num.kif1>,<num.kif2>)
Beállítja a "PCOL()" és "PROW()" értékeket. Nem ad vissza értéket.

14.8.44 STRTRAN(<kar.kif1>,<kar.kif2>[,<kar.kif3>]
[,<num.kif1>][,<num.kif2>])
Az "kar.kif1"-ben megkeresi a "kar.kif2"-t, és "kar.kif3"-ra cseréli.

A fejlesztés fontos állomása, hogy a nyomkövető szolgáltatásai is bővültek, és használatuk jelentősen egyszerűsödött.

Új funkciók:

- Az aktív adatbázisok szerkezetének bemutatása
- Az indexek számának és a kulcskifejezések kiírása
- A kapcsolódó adatbázisok, és az összerendelő kifejezések tartalmának lekérdezhetősége

A nyomkövető menüzött formában, egymásba ágyazott menük, és tájékoztató feliratok, valamint szövegorientált help-képernyők segítségével igen kényelmessé teszi a hibakeresést.

A dBASE III PLUS és a Clipper közötti kompatibilitási problémák megoldása is újabb fejlesztéseket eredményezett:

1.) dBASE III PLUS kompatibilis indexek használata

Az NDX.OBJ file lehetővé teszi .NDX kiterjesztésű file-ok létrehozását,és használatát a Clipperben. Az NDX.OBJ file-t szerkesszük hozzá a föprogramhoz, és ettől fogva a végrehajtható (.EXE) modul a .NDX index file-okat kezeli. Figyeljünk arra, hogy ugyanabból a programból egyszerre nem használhatók különböző kiterjesztésű index file-ok.

2.) A Clipper Utility programjai

A CREATE és az INDEX programokat egy DBU nevű program foglalja magába, melynek szolgáltatásai túlterjednek az előző két programén.

Lehetővé teszi az adatbázis file-ok létrehozását, az adatok módosítását, megtekintését, és indexelését is. A DBU program forrásnyelven van jelen a programcsomagban,

így le kell fordítani és szerkeszteni a MAKEDBU.BAT file segítségével. A DBU.EXE fejlett menü- és help-rendszerrel segíti az adatbázisok kezelését.

A REPORT és LABEL programokat is összevonva találjuk az RL programban. A fentiekhez hasonlóan fordítani és szerkeszteni kell a MAKERL.BAT file segítségével, és így jól használható, menüzött és help-képernyőkkel ellátott eszközhöz jutunk a formátum file-ok kezelésében.

Uj Utility program a MAKE.EXE, amely a forrás- és object file-ok összefüggéseinek leírására alkalmas a Clipper fordító számára, és folyamatossá tehető segítségével a fordítás és szerkesztés művelete.

14.9 Példa

Törzsadatok karbantartása és listázása többfelhasználós környezetben.

(A többfelhasználós funkciók nagy része megtalálható a LOCKS.PRG programban.)

A program a következő adatbázist használja:

SZEMFIL.DBF

Field	Field Name	Type	Width	Dec
1	SZAM	Character	11	
2	NEV	Character	30	
3	LAK	Character	30	
4	MUN	Character	30	
5	TEL	Character	7	
6	BER	Numeric	5	

A személyi adatokat tartalmazo file karbantartására, lekérdezésére, és kinyomtatására alkalmas a program. Ezek a funkciók egy főmenüből választhatók ki. A karbantartás funkcióit (felvitel, módosítás, törlés) újabb menübe foglaltuk.

A törzsfile-t személyi számra indexeltük, így a dolgozók ez alapján azonosíthatók. A személyi szám kezdetleges ellenőrzését is beépítettük a programba a biztosan hibás adatok bevitelének elkerülésére.

Hálózati alkalmazást feltételezve a program kezeli a zárolások esetleges ütközését is. Adatbevitelkor egy képernyő kitöltése után mindíg meg kell erősíteni az adatok helyességét ahhoz, hogy a törzsfile-ba belekerüljenek.

Lekérdezés és listázás alkalmával mód nyílik a törzsfile megadott részének leválogatására. A kiválasztási szempontok – egy logikai kifejezés formájában – a programnak megadhatók. Ehhez segítségül felsorolja a törzsfile mezőinek nevét. A kiválasztási szempont természtesen üresen is hagyható, ekkor az egész file-t kérdezzük le, illetve nyomtatjuk. Lekérdezéskor egyszerre egy dolgozó adatait látjuk, és a PgUp , valamint PgDn billentyűkkel lehet köztük lapozni.

A programokból bármikor kiléphetünk az Esc billentyű megnyomásával, és visszatérhetünk a legközelebbi menübe.

A hibaüzenetek egy dobozban jelennek meg hagjelzés kíséretében, és mindaddig a képernyőn maradnak, amíg egy billentyűt be nem ütünk.

```
**********************************
*                                *
*       CLIPPER MINTAPELDA       *
*                                *
**********************************
*                                *
*  TOBBFELHASZNALOS SZEMELYI     *
*       NYILVANTARTAS            *
*                                *
**********************************

clear screen
public sel,wsz,wne,wmh,wla,wte,wber

* A LEHETSEGES KEPERNYOK SZINEINEK BEALLITASA

if iscolor()    && HA VAN SZINES GRAFIKUS KARTYA A GEPBEN

   norm="g/n,n/gr"
   piro="r/n"
   vill="gr*/r"
   men1="gr+/b"
   men2="gr+/r,r/gr"
   men3="bg/b"
   men4="g*/n,n/gr"
   men5="gr+/r"

else   && HA NINCS SZINES GRAFIKUS KARTYA

   norm="w/n,n/w"
   piro="w+/n"
   vill="w*/n"
   men1="w+/n,n/w"
   men2=men1
   men3=norm
   men4=vill
   men5=piro

endif

set confirm on          && MEZO VEGEN <ENTER> SZUKSEGES
set date ansi           && A DATUM FORMATUMA EE.HH.NN
set deleted on          && A TOROLT REKORDOKAT NEM ADJA AT
set exclusive off       && AZ ALLOMANYOKAT OSZTOTTAN NYITJA
set scoreboard off      && NEM IRJA KI A BILLENTYUZET ALLAPOTAT
set wrap on             && A MENUZES CIKLIKUS
set century on          && A DATUM EEEE.HH.NN FORMATUMU

* MEGRPOBALJA OSZTOTTAN MEGNYITNI A FILE-T

if .not. nuse("szemfil",.f.,10)
   do doboz with 10,15,"A szemelyi file-t masvalaki","kizarolagos joggal",;
                  "hasznalja!"
   clear screen

    * NEM SIKERULT...
```

```
   return
endif

* AZ INDEX ALLOMANY MEGNYITASA

set index to szemfil

flt=space(76)
do keret
sel=0

* FOMENU KIIRASA

do while sel#4
   set color to &men3
   do menusc with 4
   @ 4,30 say "F O M E N U"
   @ row()+2,13 say "Karbantartas"
   @ row()+2,13 say "Lekerdezes"
   @ row()+2,13 say "Listazas"
   @ row()+2,13 say "A munka befejezese"
   set color to &men2
   menu to sel
   if sel#0 .and. sel#4
      pr="pro"+str(sel,1)

      do &pr             && A KIVALASZTOTT MENUPONT HIVASA

      sel=0
   endif
enddo
clear screen
return

* ALMENU1

procedure pro1
   sel=0
   do while sel#4
      set color to &men3
      do menusc with 4
      @ 4,30 say "Karbantartas"
      @ row()+2,13 say "Uj dolgozo felvitele"
      @ row()+2,13 say "Dolgozo adatainak modositasa"
      @ row()+2,13 say "Dolgozo adatainak torlese"
      @ row()+2,13 say "Vissza a fomenure"
      set color to &men2
      menu to sel
      if sel#0 .and. sel#4
         pr="pro1"+str(sel,1)

         do &pr          && A KIVALASZTOTT MENUPONT HIVASA

         sel=0
      endif
   enddo
return
```

```
* UJ DOLGOZO FELVITELE

procedure pro11
   set color to &norm
   do ernyo with "Uj dolgozo felvitele"   && AZ INPUT URES ERNYO KIIRASA
   val=1
   wsz="@"

   do while wsz#space(11)          && ADDIG, MIG A SZEMELYI SZAM NEM URES

      if val=1                     && HA AZ ELOZO BEVITEL RENDBEN,
         do init                   && TOROLJE A MUNKAVALTOZOKAT
      endif

      do szambe                    && AZ INPUT MEZOK MEGJELENITESE
      do beolv
      clear gets

      @ 23,2 say "Vissza a menuhoz, ha a szemelyi szam=[          ]"

      do szambe                    && SZEMELYI SZAM BEOLVASASA, ELLENORZESE
      read

      @ 23,2 clear to 22,78

      if lastkey()=27              && HA <ESC> -ET NYOMTAK VISSZA A MENUHOZ
         return
      endif

      if wsz#space(11)
         seek wsz                  && VAN MAR ILYEN DOLGOZO ?

         if found()                && VAN
            do attolt              && DOLGOZO ADATAIT KIIRJA
            do beolv
            clear gets
            do doboz with 20,10,"Ilyen dolgozo mar van!"
            val=1

         else                      && NINCS
            do beolv
            read
            val=2
            do rendben with val

            if val=1               && ADATBEVITEL RENDBEN

               if .not. add(10)

                  * SIKERTELEN FILE-BOVITES, HIBAUZENET

                  do doboz with 15,10,"A szemelyi allomanyt valaki",;
                                      "kizarolagos joggal","hasznalja!"
                  val=2

               else
```

```
                    *SIKERES FILE-BOVITES, ADATOK BEIRASA

                    replace szam with wsz,nev with wne,lak with wla;
                            mun with wmh,tel with wte,ber with wber
                    unlock

                endif
             endif
          endif
       endif
    enddo
return

* MODOSITAS

procedure pro12
    set color to &norm
    do ernyo with "Dolgozo adatainak modositasa"  && INPUT ERNYO
    val=1
    wsz="@"

    do while wsz#space(11)          && ADDIG, MIG A SZEMELYI SZAM NEM URES

       if val=1                     && HA AZ ELOZO BEVITEL RENDBEN,
          do init                   && TOROLJE A MUNKAVALTOZOKAT
       endif

       do szambe                    && AZ INPUT MEZOK MEGJELENITESE
       do beolv
       clear gets

       @ 23,2 say "Vissza a menuhoz, ha a szemelyi szam=[          ]"

       do szambe                    && SZEMELYI SZAM BEOLVASASA, ELLENORZESE
       read

       @ 23,2 clear to 22,78

       if lastkey()=27              && HA <ESC> -ET NYOMTAK VISSZA A MENUHOZ
          return
       endif

       if wsz#space(11)
          seek wsz                  && VAN MAR ILYEN DOLGOZO ?

          if .not. found()          && NINCS
             do doboz with 20,10,"Ilyen dolgozo meg nincs!"
             val=1

          else                      && VAN
             do attolt
             do beolv
             read
             val=2
             do rendben with val

             if val=1               && ADATBEVITEL RENDBEN
```

```
            if .not. rlc(10)

               * SIKERTELEN REKORD-ZAROLAS, HIBAUZENET

               do doboz with 15,10,"A szemelyi allomanyt valaki",;
               "kizarolagos joggal",;
               "hasznalja!"
               val=2

            else

               * SIKERES REKORD-ZAROLAS, ADATOK BEIRASA

               replace nev with wne,lak with wla;
                       mun with wmh,tel with wte,ber with wber
               unlock

            endif
         endif
      endif
   endif
enddo
return

* TORLES

procedure pro13
   set color to &norm
   do ernyo with "Dolgozo adatainak torlese"   && INPUT ERNYO
   val=1
   wsz="@"

   do while wsz#space(11)          && ADDIG, MIG A SZEMELYI SZAM NEM URES

      if val=1                     && HA AZ ELOZO BEVITEL RENDBEN,
         do init                   && TOROLJE A MUNKAVALTOZOKAT
      endif

      do szambe                    && AZ INPUT MEZOK MEGJELENITESE
      do beolv
      clear gets

      @ 23,2 say "Vissza a menuhoz, ha a szemelyi szam=[          ]"

      do szambe                    && SZEMELYI SZAM BEOLVASASA, ELLENORZESE
      read

      @ 23,2 clear to 22,78

      if lastkey()=27              && HA <ESC> -ET NYOMTAK VISSZA A MENUHOZ
         return
      endif

      if wsz#space(11)
         seek wsz                  && VAN MAR ILYEN DOLGOZO ?

         if .not. found()          && NINCS
```

```
            do doboz with 20,10,"Ilyen dolgozo meg nincs!"
            val=1

         else                        && VAN
            do attolt
            do beolv
            clear gets
            val=2
            do rendben with val   && ADATBEVITEL RENDBEN

            if val=1

               if .not. rlc(10)

                  * SIKERTELEN REKORD-ZAROLAS, HIBAUZENET

                  do doboz with 15,10,"A szemelyi allomanyt valaki",;
                  "kizarolagos joggal",;
                  "hasznalja!"
                  val=2

               else

                  * SIKERES REKORD-ZAROLAS, TORLES

                  delete
                  unlock

               endif
            endif
         endif
      endif
   enddo
return

* LEKERDEZES

procedure pro2
   flt=space(76)
   set color to &norm
   do ernyo with "Dolgozok adatainak lekerdezese"  && INPUT ERNYO

   for i=1 to fcount()
      @ 5+i*2,19 say fieldname(i)   && MEZONEVEK KIIRASA
   next

   do while .t.

      * AZ ADATOK EGY RESZE, VAGY EGESZE LEKERDEZHETO

      @ 20,3 say "Adja meg a kivalasztasi szempontokat!"
      @ 21,3 get flt
      read
      wk=trim(flt)

      if wk==""  && HA NINCS KIV. SZEMPONT, AZ OSSZES ADAT KIIRODIK
         exit
```

```
      endif

      if type("&wk")#"L"  && HIBAS KIFEJEZES

         do doboz with 10,10,"Hibas a szelekcios szempont!"

         if lastkey()=27  && HA <ESC> -ET NYOMTAK, VISSZA A MENUHOZ
            return
         endif

         loop
      endif

      exit
   enddo

   set filter to &flt     && ADATOK LEVALOGATASA
   go top

   if eof()               && NINCS MEGFELELO REKORD

      do doboz with 7,40,"Nincs a felteteleknek megfelelo","dolgozo!"
      return
   endif

   do while lastkey()#27  && HA NEM <ESC> -ET NYOMTAK
      do attolt           && ADATOK KIIRASA
      @ 7,19 get wsz
      do beolv
      clear gets
      keyboard ""
      @ 22,1 say ""
      wait chr(186)+" Kovetkezo=Pg Dn, elozo=Pg Up, menure=Esc"

      do case
         case lastkey()=18  && HA <Pg Up> -T NYOMTAK
            skip -1
            if bof()
               go bottom
            endif

         case lastkey()=3   && HA <Pg Dn> -T NYOMTAK
            skip 1
            if eof()
               go top
            endif

         case lastkey()=27  && HA <ESC> -ET NYOMTAK

         otherwise          && HA EZEKEN KIVUL MAST NYOMTAK

            do doboz with 7,40,"A billentyu nem definialt!"

      endcase
   enddo
   set filter to
return
```

```
* LISTAZAS

procedure pro3
   flt=space(76)
   set color to &norm
   do ernyo with "Dolgozok adatainak listazasa"  && INPUT ERNYO

   for i=1 to fcount()
      @ 5+i*2,19 say fieldname(i)  && MEZONEVEK KIIRASA
   next

   do while .t.

      * AZ ADATOK EGY RESZE, VAGY EGESZE LEKERDEZHETO

      @ 20,3 say "Adja meg a kivalasztasi szempontokat!"
      @ 21,3 get flt
      read
      wk=trim(flt)

      if wk==""  && HA NINCS KIV. SZEMPONT, AZ OSSZES ADAT KIIRODIK
         exit
      endif

      if type("&wk")#"L"  && HIBAS KIFEJEZES

         do doboz with 10,10,"Hibas a szelekcios szempont!"

         if lastkey()=27  && HA <ESC> -ET NYOMTAK, VISSZA A MENUHOZ
            return
         endif

         loop
      endif

      exit
   enddo

   set filter to &flt     && ADATOK LEVALOGATASA
   go top

   if eof()               && NINCS MEGFELELO REKORD

      do doboz with 7,40,"Nincs a felteteleknek megfelelo","dolgozo!"
      return
   endif

   do while .not. isprinter()  && NINCS NYOMTATO

      do doboz with 10,10,"Kerem, kapcsolja be a nyomtatot!"
      if lastkey()=27
         return
      endif
   enddo

   set device to printer
   setprc(66,0)
```

```
   do while .not. eof()  && NYOMTATAS

      if prow()>60       && LAPEMELES, FEJLEC

         @ 0,15 say "Dolgozok adatainak listaja"
         @ 2,0 say "Szem. szam    Nev"
         @ 2,50 say "Ber"
         @ 3,0 say replicate("=",53)
      endif

      @ prow()+1,0 say szam
      @ prow(),14 say nev
      @ prow(),48 say ber picture "99999"
      skip
   enddo

   set device to screen
return

* KEPERNYO-RAJZOLO ELJARAS

procedure ernyo
   parameters cim

   do keret  && KERET-RAJZOLAS

   @ 4,int((80-len(cim))/2) say cim  && CIM KOZEPRE

   @ 7,3 say  "Szemelyi szam :"
   @ 9,3 say  "Nev           :"
   @ 11,3 say "Lakcim        :"
   @ 13,3 say "Munkahely     :"
   @ 15,3 say "Telefonszam   :"
   @ 17,3 say "Fizetes       :"
return

* SZEMELYI SZAMOT BEOLVASO ELJARAS

procedure szambe
   @ 7,19 get wsz pict "99999999999" valid ellenor(wsz)
return

* DOLGOZO ADATAIT BEOLVASO ELJARAS

procedure beolv
   @ 9,19 get wne pict "@!"
   @ 11,19 get wla pict "@!"
   @ 13,19 get wmh pict "@!"
   @ 15,19 get wte pict "999-999"
   @ 17,19 get wber pict "99999" valid wber>0
return

* MUNKAVALTOZOKAT TORLO ELJARAS

procedure init
   wsz=space(11)
   wne=space(30)
```

```
   wmh=space(30)
   wla=space(30)
   wte=space(7)
   wber=0
return

* MUNKAVALTOZOKAT ADATBAZISBOL FELTOLTO ELJARAS

procedure attolt
   wsz=szam
   wne=nev
   wmh=mun
   wla=lak
   wte=tel
   wber=ber
return

* SZEMELYI SZAMOT ELLENORZO FUGGVENY

function ellenor
   parameters mit
   private summa

   if mit=space(11)  && SZEM. SZAM URES
      return .t.
   endif

   summa=0

   for i=1 to 10
      summa=summa+i*val(substr(mit,i,1))
   next

   if summa%11#val(substr(mit,11,1))  && AZ ELLNORZO JEGY NEM MEGFELELO

      do doboz with 12,10,"Hibas a szemelyi szam!"
      return .f.
   endif

return .t.

* KERET-RAJZOLO ELJARAS

procedure keret
  @ 0,0 say chr(201)+replicate(chr(205),78)+chr(187)
  @ 1,0 say chr(186)+replicate(" ",78)+chr(186)
  @ 2,0 say chr(204)+replicate(chr(205),78)+chr(185)
  @ 3,0 say chr(186)+replicate(" ",78)+chr(186)
  for i=4 to 23
     @ i,0 say chr(186)+replicate(" ",78)+chr(186)
  next
  @ 22,0 say chr(204)+replicate(chr(205),78)+chr(185)
  @ 24,0 say chr(200)+replicate(chr(205),78)+chr(188)
  @ 1,30 say "Clipper mintapelda"
  @ 2,66 say date()
return
```

```
* MENU-RAJZOLO ELJARAS

procedure menusc
parameters ig

  do keret  && KERET MEGRAJZOLASA

  @ 23,9 say "Fol, le :     Kivalasztas:"    && UZENETSOR
  set color to &men5

  for i=1 to ig-1
     @ 6+2*(i-1),9 prompt " "+str(i,1)+" "   && MENUPONTOK
  next
  @ row()+2,9 prompt " 0 "

  set color to &vill                          && UZENETSOR
  @ 23,18 say "+ +"
  @ 23,36 say "[Return]"
  set color to &men1

return

* HIBAUZENETEK ELJARAS

procedure doboz
  parameters sor,osz,sz1,sz2,sz3

  ?? chr(7)  && HANGJELZES

  private l1,l2,l3,l,m,w,a
  set color to &piro

  save screen
  a=' '

  * DOBOZ MERETENEK MEGHATAROZASA

  l1=len(sz1)
  l=l1

  if pcount()>3       && HA A SZOVEG KET VAGY HAROM SOROS
    l2=len(sz2)
    l=if(l1>l2,l1,l2)

    if pcount()>4     && HA A SZOVEG HAROM SOROS
      l3=len(sz3)
      l=if(l3>l,l3,l)
    endif

  endif

  dz=chr(201)+chr(205)+chr(187)+chr(186)+chr(188)+chr(205)+chr(200)+chr(186)+;
  chr(176)
  @ sor,osz,sor+pcount()+1,osz+l+4 box dz    && DOBOZT RAJZOL

  * SZOVEG(EK) BEIRASA
```

```
  l1=int((l-l1)/2)
  @ sor+2,osz+2+l1 say sz1

  if pcount()>3                    && HA A SZOVEG KET VAGY HAROM SOROS
    l2=int((l-l2)/2)
    @ sor+3,osz+2+l2 say sz2

    if pcount()>4                  && HA A SZOVEG HAROM SOROS
      l3=int((l-l3)/2)
      @ sor+4,osz+2+l3 say sz3
    endif

  endif

  keyboard ""     && KITORLI A BILLENTYUZET-PUFFERT
  wait ""         && VAR EGY BILLENTYU MEGNYOMASARA
  restore screen

  ?? chr(7)  && HANGJELZES

  set color to &norm
return

* ADATBEVITELT MEGEROSITO ELJARAS

procedure rendben
parameters v
  save screen

  @ 23,0 say chr(186)+replicate(" ",11)+"Rendben ?"+replicate(" ",58)+chr(186)

  set color to &men4

  @ 23,7 say "->"                  && UZENETSOR

  set color to &norm

  @ 23,23 prompt " Igen "          && UZENETSOR
  @ 23,29 prompt " Nem "           && UZENETSOR

  menu to v
  restore screen
return

* FILE-BOVITO ELJARAS

function add
  parameters wait
  private forever

  append blank  && LETREHOZ EGY URES REKORDOT

  if .not. neterr()  && SIKERES LETREHOZAS
     return (.t.)
  endif

  forever = (wait = 0)
```

```
  do while (forever .or. wait > 0)  && SIKERTELEN KISERLET ESETEN
     append blank                   && MEGADOTT IDEIG PROBALJA UJRA
     if .not. neterr()
        return .T.    && KOZBEN SIKERULT...
     endif
     inkey(.5)
     wait = wait - .5
  enddo

return (.f.)          && EGYALTALAN NEM SIKERULT...

* REKORD-ZAROLO ELJARAS

function rlc
  parameters wait
  private forever

  if rlock()          && SIKERES ZAROLAS
     return (.t.)
  endif

  forever = (wait = 0)
  do while (forever .or. wait > 0)  && SIKERTELEN KISERLET ESETEN
     if rlock()                     && MEGADOTT IDEIG PROBALJA UJRA
        return .T.    && KOZBEN SIKERULT...
     endif
     inkey(.5)
     wait = wait - .5
  enddo
return (.f.)          && EGYALTALAN NEM SIKERULT...

* HALOZATI MEGNYITO ELJARAS

function nuse
  parameters file, ex_use, wait
  private forever

  forever = (wait = 0)
  do while (forever .or. wait > 0)   && A FILE MEGNYITASAVAL MEGADOTT
                                     && IDEIG PROBALKOZIK

     if ex_use                       && MEGNYITAS KIZAROLAGOS HASZNALATRA
        use &file exclusive

     else                            && MEGNYITAS KOZOS HASZNALATRA
        use &file

     endif

     if .not. neterr()               && SIKERES MEGNYITAS
        return (.t.)
     endif

     inkey(1)
     wait = wait - 1
  enddo
return (.f.)                         && SIKERTELEN MEGNYITAS
```

15. A FoxBase+ fejlesztő környezet

15.1 A FoxBase+ rendszer sajátosságai

15.1.1 Bevezetés

A FoxBase+ programrendszer a dBASE III PLUS jellegű rendszerek közül a leggyorsabb. A Communication of the ACM 1986 márciusi számában megjelent összehasonlítás alapján 6.74-szer gyorsabb az eredeti rendszernél, s 2.3-szer a Clippernél. Természetesen az azóta eltelt időszak alatt mindegyik gyártó tovább fejlesztette a saját termékét, de a tapasztalatok alapján ezek a különbségek nem csökkentek igazán.

A FoxBase+ rendszer – eltérően a többi dBASE követőtől – nem csak fordítót, hanem interpretert is tartalmaz, amelyik lehetővé teszi az elkészített programok tesztelését, s interaktiv környezetben való javítását is.

Jelen rövid összefoglalás a FoxBase+ 1.21-es (1987 januári) változatát ismerteti. A rendszeren azóta is folyamatosan végeznek kisebb-nagyobb módosításokat, de ez annak legfontosabb tulajdonságait nem érinti.

A FoxBase+ és a dBASE közötti kompatibilitás az alábbi összetevőkre terjed ki:

- program file-ok (.PRG)
- adat file-ok (.DBF)
- memo file-ok (.DBT)
- memóriaváltozók (.MEM)
- képernyőformátum file-ok (.FMT)
- jelentés formátum file-ok (.FRM)
- cimke definiciós file-ok (.LBL)

A FoxBase+ a dBASE III PLUS-tól eltérő algoritmusú indexelést használ. Minden olyan esetben, amikor olyan indexet használunk, aminek csak dBASE-beli megfelelője létezik (.NDX kiterjesztéssel), akkor a FoxBase+ az újfajta indexet automatikusan létrehozza, de .IDX kiterjesztéssel.

15.1.2 FoxBase+ bővítések

A Fox Software cég a dBASE III PLUS-szal való teljes mértékű kompatibilitásra törekedett, amit sikerült is megvalósítani. Az interpeter esetében ez a fejlesztői környezetre is vonatkozik, nemcsak a programnyelvre. Néhány ponton azonban a FoxBase+ lényeges bővítéseket is tartalmaz.

15.1.2.1 Tömbváltozók használata

A FoxBase+ lehetővé teszi egy- vagy kétdimenziós memória változótömbök használatát. A tömbindexek 1-gyel kezdődnek és egész számok. A DIMENSION X(2,3) utasítás például 2x3-as tömböt definiál. Ezt követően a tömb elemeire tetszőleges kifejezésekben hivatkozhatunk.

A FoxBase+ 2.0-ás és későbbi verziói két további parancsot tartalmaznak, amelyek a mezők és a tömbök közötti adatátvitelt segítik. Ezek a következők:

GATHER FROM <tömb> [FIELDS <mezőlista>]
SCATTER [FIELDS <mezőlista>] TO <tömb>

A GATHER parancs segítségével a tömb elemét az összes (vagy a megnevezett) mezőkbe másolhatjuk, míg a SCATTER parancs ennek az ellenkezőjét végzi. A MEMO típusú mezők nem vesznek részt a másolásban.

Ha a tömb elemeinek és a mezőknek a száma eltérő, akkor a felesleges elemek nem másolódnak át, illetve a felesleges elemekbe nem másolódik új érték. A GATHER parancs esetén a FoxBase+ típus-ellenőrzést is végez. Ha a tömb elemének típusa nem egyezik a mező típusával, akkor "DATA TYPE MISMATCH" hibaüzenetet kapunk.

15.1.2.2 Többszörös RELATION használata

A SET RELATION TO utasítás szintaxisa egy további, az ADDITION kulcsszóval bővült, amelynek hatására az utasítás nem törli az előzőleg felépített relációkat.

15.1.2.3 Rendszerfüggvények

A FoxBase+ egy új SYS függvényt tartalmaz, amelyik lehetővét teszi különböző, a rendszer (értsd alatta: a DOS) konfigurálásakor beállított értékek lekérdezését.

15.1.2.4 Eljárások száma az eljárásfile-ban

A FoxBase+ 128 eljárás definiálását teszi lehetővé egyetlen file-ban.

15.1.2.5 Eljárásfile-ok létrehozása

A FoxBase+ tartalmaz egy FOXBIND elnevezésű segédprogramot, amelyik lehetővé teszi különböző programfile-ok egyetlen eljárásfile-ba való szerkesztését.

15.1.2.6 dBASE II file-kompatibilitás

A FoxBase+ képes a USE paranccsal tetszőleges dBASE II vagy FoxBase file használatára. Ennek ellenére célszerű a COPY dBASE III PLUS utasítással egy új, most már dBASE III PLUS kompatibilis file-t létrehozni, s azt használni.

15.1.2.7 MIN/MAX függvények a dátumokon

Lehetőség van a MIN és MAX függvények használatára dátum típusu értékekre is.

15.1.2.8 DTOC függvény módosítása

A DTOC függvény tartalmaz egy további argumentumot, amelyik a dátum értékét a rendezésekben (INDEX) használható alakban adja vissza.

15.1.2.9 BROWSE utasítás módosítása

A BROWSE parancs esetén is lehetőség van memo mezők szerkesztésére a <CTRL-PgDn> billentyű lenyomásával. A szerkesztést a <CTRL-PgUp> billentyű lenyomásával fejezhetjük be.

A BROWSE menüben a SKIP +/-n parancsot is használhatjuk.

15.1.2.10 MODIFY COMMAND utasítás módosítása

A szerkesztést két további lehetőség könnyíti meg. A <CTRL-K>T és a <CTRL-K>E billentyűzés a szöveg elejére, illetve a végére állítja a kurzort.

A FoxBase + esetében a szerkesztendő file nagyságát csak a CONFIG.DB file-ban megadott MAXMEM változó értéke befolyásolja.

15.1.2.11 Nyitott file-ok száma

A FoxBase + rendszer használatakor a következő számú file-ok lehetnek egyidejűleg nyitva:

- Összesen maximum 48 file lehet megnyitva.
- 10 adatbázisfile lehet egyidejüleg megnyitva (az esetleg hozzájuk tartozó memo file-okkal).
- Legfeljebb 21 indexfile lehet nyitva.

Természetesen a CONFIG.SYS file-ban a FILES= paramétert legalább 10-zel nagyobbra kell állítani a FoxBase + rendszerben szereplő file-ok számánál.

15.1.2.12 Nyomtató időtúllépése

A FoxBase + lehetővé teszi annak a megadását, hogy a FoxBase + hányszor kíséreljen egy karaktert a nyomtatóra elküldeni, mielőtt "Printer not ready" (Nem áll rendelkezésre a nyomtató) hibaüzenetet küld. Ezt a CONFIG.DB file-ban a TIME paraméter értékének megadásával állíthatjuk be. Kezdeti értéke 6000, ez kb. 6-10 másodpercnek felel meg.

15.1.2.13. Felhasználói függvények

A FoxBase + 2.0-ás és későbbi verziói lehetővé teszik saját, ún. felhasználó által definiált (User Defined Function) függvények használatát. A függvényt tetszőleges program definiáltatja. Magát a programot - hasonlóan a 'normális' programokhoz - eljárásfile-ban is elhelyezhetjük, s mind eredeti, mind fordított változatát használhatjuk.

Magára a függvényre a program vagy az eljárásfile moduljának a nevével lehet hivatkozni. Ezt követik a függvény paraméterei kerek zárójelek közt, vesszővel elválasztva:

<név> (<aktuális paraméter-lista>)

A zárójelpár akkor sem maradhat el, ha nincs egyetlen paraméter sem.

Ha a függvénynek egy vagy több paramétere van, akkor a függvényt definiáló program első utasítása

PARAMETER < paraméter-lista >

alakú kell hogy legyen. A < paraméter-lista > elemei a programban használt memória-változók nevei. A függvény értékének visszaadására a RETURN parancs szolgál:

RETURN < kifejezés >

Ez a parancs felfüggeszti a függvény kiértékelését és a < kifejezés > aktuális értékével tér vissza. A függvény értékét az alábbi parancsokban használhatjuk:

- STORE
- REPLACE
- vezérlő utasítások (IF, WHILE, CASE, stb.)
- ? és ??
- DISP, LIST
- @...GET...VALID

A felhasználó által készített függvényeket definiáló programok nem változtathatják meg az adatbázis állapotát, mert akkor nem kívánt mellékhatások léphetnek fel.

Példa

Tegyük fel, hogy a KOB.PRG file az alábbi sorokat tartalmazza:

```
** KOB.PRG **
PARAMETER u
RETURN u*u*u
** END **
```

Erre a függvényre pl. a következőképpen hivatkozhatunk:

- ? KOB(3) + 2
- STORE KOB(4) + KOB(2) TO Y

15.1.3 Nem támogatott elemek

A FoxBase + nem tartalmazza a következő utasításokat és lehetőségeket:

- ASSIST
- CREATE/MODIFY/SET VIEW
- CREATE/MODIFY/SET QUERY
- CREATE/MODIFY SCREEN
- A katalógusra vonatkozó műveletek
- IMPORT/EXPORT
- A .DIF, .SLK és .WKS file-ok transzformálása

15.1.4 A FoxBase + hardware igénye és technikai adatai

A FoxBase + használatához 375 kByte szabad memóriára és legalább két hajlékonylemez egységre van szükség. Legalább egy, megfelelő kapacitású merevelemezes egység használata mindenképp ajánlott.

Aritmetikai coprocesszor használata észrevehetően meggyorsítja a programrendszer működését.

A FoxBase + a következő korlátokkal képes az adatbázis-file-okat feldolgozni:

Az adatbázis rekordjai	1.000.000.000
Karakter/rekord	4.000
Mező/rekord	128
Numerikus adat pontossága	16
Karakterlánc hossza	254
Parancssor hossza	254
Indexkulcs hossza	100
Egyidejüleg megnyitható file-ok	48

A FoxBase + működtetéséhez az MS/PC-DOS legalább 2.0-ás verziója szükséges.

15.2 A FoxBase + programrendszer felépítése

15.2.1 A fordító

A FoxBase + programrendszer legértékesebb darabja. Segítségével .FOX kiterjesztésű object-file-t hozhatunk létre. Ezek a file-ok közvetlenül az interpeterből vagy a futtató rendszerből indíthatók. A FoxBase + fordítója – szemben a Clipper-rel – nem hoz létre önállóan futtatható (.EXE) kiterjesztésű file-t!

A fordító a

FOXPCOMP <file1> [<file2>...<fileN>]

paranccsal indítható el. A <file1>,...,<fileN> file-azonosítók a DOS globális filenév karaktereit (* és ?) is tartalmazhatják. Minden egyes program önállóan kerül lefordításra, így N darab új file jön létre, amelyik a lefordított programokat tartalmazza.

A file nevekben a kiterjesztést meg kell adni. Ennek oka, hogy a FOXPCOMP mind program, mind formátum file-ok fordítására alkalmas. A lefordított programot tartalmazó file kiterjesztése .FOX, míg a formátum file fordításáé .FMX lesz.

A FoxBase + interpeter mindig a lefordított programot futtatja (ha létezik fordított változat). Éppen ezért, ha a MODIFY COMMAND paranccsal módosítottunk egy file-t, akkor az esetleg ugyanabban a katalógusban levő, fordított (.FOX) változata automatikusan törlődik!

15.2.2 Eljárás-file generálása

Az önálló program file-okban levő programok egyetlen eljárás file-ba való tömörítése, ha a futási időt nem is, de a tárba töltés idejét nagymértékben csökkenti.

A fenti feladat egyszerű elvégzésere a FoxBase+ rendszer tartalmazza a FOXBIND segédprogramot, amelyik az input-jaként megadott programokból létrehoz egy eljárásfile-t. Minden egyes programfile-t bemásol az eljárásfile-ba egy

PROCEDURE <programnev>

sorral megelőzve.

A segédprogram (a DOS-ból) a következő parancs kiadásával indítható el:

FOXBIND <eljárásfile> <file1> [<file2>...<fileN>]

A segédprogram a megadott file-okból egyetlen file-t állít elő, ennek lesz a neve <eljárásfile>. A <file1>,...,<fileN> file-azonosítók tartalmazhatnak globális filenév karaktereket (* és ?) is. A kiterjesztések megadása kötelező.

Ha az <eljárásfile> már létezik, akkor a segédprogram a létező file-hoz fűzi hozzá a megadott eljárásokat.

15.2.3 A FoxBase+ interpretere

Meglevő programok (akár eredeti, akár fordítottak) futtatása vagy az interpeter vagy a futtató rendszer segítségével történhet. Az interpreter elindításához a

FOXPLUS vagy MFOXPLUS

parancsok egyikét kell a DOS szintjén kiadni. (A második a program többfelhasználós változata). Ennek hatására megjelenik a FoxBase+ logoja és a rendszer készen áll a parancsaink elvégzésére.

Lehetőség van automatikus programindításra is. Ekkor a

FOXPLUS <programnév>
MFOXPLUS <programnév>

parancsok egyikét kell a DOS szintjén kiadni. Ennek hatására elindul a FoxBase+ interpeter, ami automatkusan betölti a <programnev> nevű programot és elindítja. Ha a programnak létezik lefordított (azaz .FOX kiterjesztésű) változata, akkor az fog elindulni.

Az interpreter (és a futtató rendszer is) alapértelmezés szerint 64 kByte-os ún. cache memóriát tartalmaz. Ez az a memóriaterület, ahová a FoxBase+ betölti a lefordított és használt programfile-okat. Egyidőben egyszerre több program is lehet a cacheben. Ha újabb file-t aktivizálunk (például a DO paranccsal) és nincs elég hely a cache memóriában, akkor az utoljára használt file vagy file-ok törlődnek.

A FoxBase + interpreter használata gyakorlatilag megegyezik a dBASE III PLUS rendszerben a DBASE parancs kiadásának.

15.2.4 A futtató rendszer

A FoxBase + programrendszer tartalmazza a FOXPRUN (többfelhasználós változatban MFOXPRUN) programot, amelyik csak lefordított (azaz .FOX kiterjsztésű) változatok futtatására alkalmas. Ez a futtató rendszer nem kezeli az alábbi parancsokat:

- A menü vezérelt SET
- SET STEP ON
- HISTORY
- SET ECHO ON

A futtató rendszer átlapolásához (az overlay rendszerhez) két file szükséges: FOXPRUN.OVL és FOXPRUN.OV2. Az utóbbi csak az alábbi parancsok végrehajtásához szükséges:

- CREATE/MODIFY STRUCTURE
- CREATE/MODIFY REPORT
- CREATE/MODIFY LABEL

Ha programunk nem használja ezeket a parancsokat, akkor a FOXPRUN.OV2 file-t akár törölhetjük is a lemezről.

15.2.5 Képernyő kezelés

A FoxBase + (hasonlóan sok alkalmazói programhoz) kihasználja az IBM PC szabványos hardware architektúráját, s közvetlenül, a BIOS kikerülésével ír a képernyőre, ezzel nagymértékben felgyorsítva ezeket a műveleteket. Néhány 'kompatibilis' gép esetében ez azonban gondot jelenthet. A rendszer indításakor ezért megadhatjuk, hogy a program a szabvány BIOS rutinokat használja. Ehhez a parancssorban a -NOTIBM paramétert is meg kell adni, például:

```
MFOXPLUS main -NOTIBM
```

A fenti parancs hatására a tárba töltődik az MFOXPLUS majd a "main" nevű program, s végül a gyors képernyő rutinok kikapcsolódnak.

15.3 A FoxBase + rendszer konfigurálása

15.3.1 Bevezetés

Hasonlóan a dBASE (vagy akár a DOS) programrendszerhez, lehetőség van egy konfigurációs file megadására, amelyik az elindítandó (interpreter vagy futtató rendszer) program kezdeti paramétereit beállítja.

A konfigurációs file neve CONFIG.FX. Ha azonban ilyen nevű file nincs, akkor a kompatibilitás érdekében a rendszer a CONFIG.DB nevű file-t is megkeresi. Ha egyik sincs az aktuális tartalomjegyzékben, akkor az eleve beállított (default) értékekkel indul a rendszer.

A CONFIG.FX szabványos ASCII file, amely

<tétel> = <érték>

alakú sorokat tartalmaz. Szerkesztésére tetszőleges szövegszerkesztőt használhatunk.

15.3.2 Konfigurációs tételek

A tételek legtöbbje a SET parancsban megadható paraméterekkel kapcsolatos, ezeket csak a következő szakaszban (3.3) soroljuk fel. A további tételek a FoxBase+ tábláinak, puffereinek és egyéb erőforrásainak méretét adják meg, ezért a rendszer működését jobban befolyásolják, mint a SET típusú parancsok.

15.3.2.1 BUCKET

Az érték a GET, PICTURE és RANGE utasítások tárolására szolgáló memória nagyságát adja Kbyte-ban. Egyetlen GET memóriaigénye 30 és 302 byte között változhat. A paraméternek az 1-32 intervallumba kell esnie, kezdeti értéke 4.

15.3.2.2 BUFFERS

Az I/O műveletekhez rendelt pufferek számát adja meg. A teljes memóriaigényt ennek a számnak 2K-val való megszorzásával kaphatjuk meg. Azonban egészen nagy adatbázisokat leszámítva ezt az értéket 10-12-re is csökkenthetjük. A pufferek számának növelése gyorsítja a rendszer működését, de több memóriát igényel. A paraméter értékének a 4-31 intervallumba kell esnie, kezdeti értéke 31.

15.3.2.3 FILES

A FoxBase+ rendszerben egyidejűleg megnyitható file-ok számát jelenti. Egy-egy file-hoz mintegy 130 byte-ból álló terület tartozik. A paraméternek a 16-48 intervallumba kell esnie, kezdeti értéke 16.

15.3.2.4 HISTORY

A paraméter a parancs veremben tárolandó parancsok maximális számát adja meg. Lehetséges értéke a 0-16000 intervallumba esik. Kezdeti értéke 20.

15.3.2.5 HMEMORY

A paraméter a parancs verem nagyságát adja meg Kbyte-ban. Ennek nagyságát a tárolni kívánt parancsok számából és az átlagos parancshosszból lehet kiszámítani.

Minden egyes parancs tárolása további 8 byte-ot igényel. A paraméter értékének a 0-63 intervallumba kell esnie, kezdeti értéke 5.

15.3.2.6 INDEX

Az indexfile-ok kiterjesztését adhatjuk meg. Kezdeti értéke IDX. A pontot (.) nem kell beírni.

15.3.2.7 MAXMEM

A paraméter a közös memóriaterület nagyságát adja meg Kbyte-ban. Ezt használja a szerkesztő a MODIFY COMMAND parancs kiadásakor, de több parancsnak (például SORT, INDEX, REPORT) is szüksége van erre a memóriarészre. Ennek a memóriarésznek a nagysága befolyásolja, hogy a FoxBase + milyen eljárással indexszel vagy rendez. Növelése gyorsíthatja az indexelést. A paraméter értékének a 8-64 intervallumba kell esnie, kezdeti értéke 64.

15.3.2.8 MVARSIZ

A memóriaváltozók számára fenntartott memória nagysága Kbyte-ban; az 1-64 intervallumba kell esnie. Kezdeti értéke 6.

15.3.2.9 MVCOUNT

A memóriaváltozók számának maximuma. Lehetséges értéke 128-3600. Kezdeti értéke 256.

15.3.2.10 PCACHE

A lefordított programok számára fenntartott puffer terület nagysága Kbyte-ban. Ennél nagyobb lefordított programot a rendszer képtelen futtatni. Értékének a 8-64 intervallumba kell esnie, kezdeti értéke 64.

15.3.2.11 PROMPT

A FoxBase + prompt-ja. Ezzel a konfigurációs paranccsal módosítható a pont (.) prompt (készenléti jel).

15.3.2.12 TEDIT

A MODIFY COMMAND parancshoz használt külső szövegszerkesztő neve.

15.3.2.13 TIME

A FoxBase + próbálkozásainak a száma, mielőtt "Printer Not Ready" hibaüzenetet küld. Értéke 1-32767 közé kell, hogy essen. Kezdeti beállított értéke 6000 próbálkozás.

15.3.2.14 WP

A memo mezők szerkesztéséhez használt külső szövegszerkesztő neve.

15.3.3 A használható konfigurációs tételek összefoglalása

Az alábbi táblázat összefoglalja a SET parancsokból a konfigurációs file-ban is használható tételeket. A SET alapszót soghasem kell kiirni.

<tétel>	<érték>	kezdeti érték
ALTERNATE	<filenév>	
BELL	ON/OFF	ON
BUCKET	1-64	4
BUFFERS	4-31	31
CARRY	ON/OFF	OFF
CENTURY	ON/OFF	OFF
COLOR	<szín>	
COMMAND	<parancs>	
CONFIRM	ON/OFF	OFF
CONSOLE	ON/OFF	ON
DEBUG	ON/OFF	OFF
DECIMALS	0-14	2
DEFAULT	<lemez>	
DELETED	ON/OFF	OFF
DELIMITER	ON/OFF	OFF
DELIMITER	1 vagy 2 karakter	:
DEVICE	SCREEN/PRINT	SCREEN
ECHO	ON/OFF	OFF
ESCAPE	ON/OFF	ON
EXACT	ON/OFF	OFF
EXCLUSIVE	ON/OFF	ON
F<n>	"karakterek"	
FILES	16-48	16
HEADINGS	ON/OFF	ON
HELP	ON/OFF	ON
HISTORY	0-16000	20
HMEMORY	0-63	5
INDEX	kiterjesztés	IDX
INTENSITY	ON/OFF	ON
MARGIN	1-254	0
MAXMEM	8-64	64
MEMOWIDTH	8-256	50
MENU	ON/OFF	ON
MVARSIZ	1-64	6
MVCOUNT	128-3600	256
PATH	<elérési út>	

PCACHE	8-64	64
PRINT	ON/OFF	ON
PROMPT	"karakterek"	"."
SAFETY	ON/OFF	ON
SCOREBOARD	ON/OFF	ON
STATUS	ON/OFF	ON
STEP	ON/OFF	OFF
TALK	ON/OFF	ON
TEDIT	< név >	
TIME	1-32767	6000
TYPEAHEAD	ON/OFF	128
UNIQUE	ON/OFF	OFF
WP	< név >	

15.4 Többfelhasználós rendszer

15.4.1 Bevezetés

A FoxBase +, hasonlóan a dBASE-hez, rendelkezik többfelhasználós változattal is, amely lehetővé teszi, hogy IBM kompatibilis mikrogépekből álló hálózaton egyidejűleg több felhasználó ugyanazokat az adatállományokat használja.

A többfelhasználós FoxBase + kompatibilis a dBASE III PLUS Administrator programmal, az alábbi lényeges megszorításokkal:

- File titkosítás
- A PROTECT egyéb funkciói
- LIST/DISPLAY USERS

A felsorolt funkciókat a többfelhasználós FoxBase + 1.21-es verziója nem támogatja. Tervek szerint a későbbi verziók már tartalmazzák ezeket is.

15.4.2 Környezeti feltételek

A többfelhasználós FoxBase + olyan munkaállomásokon használható, amelyiken a FoxBase + maga képes üzemelni, s amelyik legalább 375 kByte szabad memóriával rendelkezik.

A hálózatnak képesnek kell lennie a standard NETBIOS funkciók végrehajtására. (Ez a munkaállomásokon legalább a DOS 3.1-es verzióját feltételezi!) A két legelterjedtebb ilyen hálózat az IBM PC Network és a Novell Advanced NetWare.

15.4.3 A többfelhasználós FoxBase + indítása

A többfelhasználós FoxBase + indítása előtt a DOS SHARE segédprogramját le kell futtatni. A SHARE-t akkor is le kell futtatni, ha önálló gépen használjuk a többfelhasználós FoxBase + -t.

A SHARE betöltése és a hálózatba való bejelentkezés után a többfelhasználós FoxBase+-t az 15.2.3 vagy az 15.2.4 pontokban mondottak szerint kell elindítani.

15.5. A SYS függvény

A SYS függvény a hálózati és egyéb környezetről ad felvilágosítást. Az argumentum értékétől függően más és más rendszerparaméterrel tér vissza. A visszaadott érték mindig karaktersorozat, még akkor is, ha ennek, mint számnak van értelme! Az egyes argumentumok jelentése a következő:

SYS(0) DOS alatt a hálózati gép számával és a munkaállomás gépének a nevével tér vissza, míg XENIX alatt a felhasználó processz azonosítójával.

SYS(1) A rendszerdátummal tér vissza az eredeti FoxBase-nek megfelelő formátummal.

SYS(2) Az éjfél óta eltelt másodpercek számával tér vissza.

SYS(3) Egy ideiglenes file nevével tér vissza, azaz egy még nem létező file nevét adja. A SYS(3) minden újabb hívása egy újabb file-nevet ad.

SYS(5) DOS alatt az aktuális meghajtó betűjelével, XENIX alatt az aktuális tartalomjegyzék nevével tér vissza. Ez az az érték, amit a SET DEFAULT TO paranccsal be lehet állítani.

SYS(6) Az aktuális PRINT egységgel tér vissza. Ez az amit a SET PRINTER TO paranccsal be lehet állítani.

SYS(7 [,mt]) Az aktuális formátum file nevével tér vissza (Amit a SET FORMAT TO paranccsal beállítottunk). Ha a munkaterület <mt> azonosítóját nem adjuk meg, akkor a kérdés az aktuális munkaterületre vonatkozik.

SYS(9) A FoxBase+ program sorozatszámát adja vissza. Ez a rendszer minden egyes példányára más és más.

SYS(10,n) A függvény az n-ben megadott napi sorszámot dátummá konvertálja. A napi sorszám az 1900 január 1-e óta eltelt napok száma.

SYS(11,d) A függvény a dátum formában megadott dátumot napi sorszámmá alakítja.

SYS(12) A még szabad memória nagyságát adja byte-okban.

SYS(13) A SET PRINT TO paranccsal beállított eszköz állapotával tér vissza. Ha az nem képes adatok fogadására, akkor a függvény értéke OFFLINE, különben READY.

SYS(14,i [,mt]) Az mt munkaterülethez i-ediknek rendelt index indexkifejezését adja. Ennek megfelelően i az 1-7, míg mt az 1-10 intervallumba kell hogy essen. Ha a szóbanforgó index nem létezik, akkor üres sztringet kapunk eredményül.

SYS(15, <tabla>, <karakterkif.>) A függvény a <karakterkif.> értékeként megadott sztring karaktereit transzformálja a <tabla> szerint. A <tabla> tetszőleges karakterkifejezés, amelynek n-edik karaktere mutatja, hogy az n ASCII kódú karaktert milyen karakterre kell cserélni.

Ezt a függvényt elsősorban rendezéseknél lehet jól használni, lehetőség van ugyanis arra, hogy az ékezetes karakterek ne az ASCII, hanem valódi helyük alapján legyenek rendezve. Ha például az ekezet nevű memóriaváltozó értéke olyan sztring, amelyikben az ékezetes betűknek megfelelő helyeken az ékezetes betűk ékezet nélküli párjai állnak, akkor például

INDEX ON SYS(15,ekezet,nev) TO inev

hatására a 'nev' mező indexelése 'normál' ABC szerint történik.

SYS(16) Az éppen futó program nevével tér vissza.

SYS(100) A SET CONSOLE kapcsoló állapotával tér vissza (ON/OFF).

SYS(101) A SET DEVICE kapcsoló állapotával tér vissza (SCREEN/PRINT).

SYS(102) A SET PRINT kapcsoló állapotával tér vissza (ON/OFF).

&; 280
@...GET; 262
@...SAY...GET; 91
ACCES(); 319
adatbázis struktúra; 20
adatbázis struktúra létrehozása; 21
adatbázis-file; 10
adatfile tartalmának megjelenítése; 52
adatfile; 17
adattitkosítás, kódolás; 291
ALIAS; 250
állapot (státusz) sor; 298
ALTERNATE; 250
APPEND FROM; 254
APPEND; 86, 92
Assist; 27
ATTRIB; 316
automatikus file zárolás; 310
azonosító; 301
bejelentkezés védelme; 291
bejelentkezési név; 300
bejelentkezési védelem; 291
belső overlay; 344
billentyűzet pufferja; 14
bővítési előjog; 294
BROWSE; 86
CHANGE; 86, 92, 311
CHR; 255
címke file; 18
címkekészítés; 71
CLOSE FORMAT; 93
CMDSET; 262
COMMAND.COM; 276
CONFIG.DB; 14
CONTINUE LOCATE; 252
CONTINUE; 89
CREATE VIEW FROM ENVIRONMENT; 90
CREATE/MODIFY REPORT; 96
CREATE/MODIFY VIEW; 90
CREATE/MODIFY; 316
.CRP kiterjesztés; 296, 307
csak olvasás (R/O) mező előjog; 294
csak olvasható – R/O (read only); 316
csoportnév; 300, 303
csoportosítás; 12
dátum változók; 25
dBASE ACCESS; 286
dBASE ADMINISTRATOR; 286
DBSYSTEM.DB; 292
DELETE; 86, 88
DELETED; 88
DISPAY/LIST TO PRINT; 95
DO; 253, 281
DO WHILE; 267
DTOC; 96
EDIT; 86, 92, 311
EJECT; 95
ENDDO; 267
ERROR(); 318
EVAL; 263
EXCLUSIVE kapcsoló; 314
explicit file-zárolás; 311
explicit rekord-zárolás; 311
EXPORT; 94
EXPRESSIONS; 253
felhasználó törlése a csoportból; 301
felhasználói hozzáférési szint; 304
felhasználói profil tárolása; 301
felhasználói profil; 299
feljegyzés file; 17
feltételek egymásba épülése; 68
file attributumok; 290
file előjog – felhasználói kategória tábla; 296
file és mező hozzáférések ellenőrzése; 291
file és mező privilégiumok tárolása; 306
file hozzáférés; 314
file hozzáférési előjogszintek; 303
file megnyitás; 314
file neve; 17
file Server; 284
file zárolás; 309
file-kiterjesztés; 17
file-műveletek; 10
file-típusok; 17
FIND; 89, 254
FLOCK(); 312, 313
fordítás; 337, 349

fordító; 338, 348
formátum-file; 39
FULL; 304
FULL előjog; 305
GO; 90
halálos zárolás; 317
hálózati teljesítőképesség; 312
HELP; 254
hibák észlelése; 318
holtpont; 317
hozzáférési szint; 291, 294, 301
IBM PC Network; 285
IMPORT; 94
INDEX; 85
indexfile létrehozása; 57
indexfile; 17, 61
indexkulcs; 61
INSERT; 86, 92
INSTALL; 288
interpretív nyelvek; 337
írható/olvasható – R/W (read/write); 316
jelszó; 300
katalógus file; 19, 76, 106
képernyőformátum létrehozása; 42
kifejezések; 11
kizárólagos megnyitás; 314
kizárólagosan megnyitott file; 310
környezeti (view) file; 73
kulcsmező; 18
külső overlay; 344
LABEL FORM TO PRINT; 95
LABEL; 96
LABEL/REPORT FORM; 254
lefordított; 338
LOAD; 259, 262
LOCATE; 89
LOCK(); 312
LOG; 256
logikai műveletek; 12
logikai változó; 25
LOGOUT; 319
lokális hálózat; 284
makro; 336, 338
matematikai műveletek; 12
megosztott módban történő megnyitás; 314
Memo field; 250
"Memo" mező; 24, 37, 55
memóriaváltozó; 10, 24
menüválasztás; 31
MESSAGE(); 318
mező hozzáférési előjogok; 304
mezőméret; 10
mezőtípusok; 23
MODIFY COMMAND; 79, 91
MODIFY LABEL/REPORT; 254
módosítási előjog; 294
műveletek elsőbbsége; 13
műveletek; 12
műveleti szabályok; 11
NDX; 256
nincs (NONE) mező előjog; 294
NONE; 304, 305
Novell Netware
Novell/Netware hozzáférési jogok; 290
numerikus pontosság; 10
olvasási előjog; 294
ON ERROR DO; 318
ON ERROR; 318
ON; 280
overlay; 341, 342, 343, 348
overlay könyvtár; 343
overlay-technika; 337, 343
PARAMETER; 268
parancssor; 10
PROTECT menüterület; 298
PROTECT.EXE; 291
QUERY; 269
R/O; 304, 305
READ; 92
RECALL; 88
rekord törlése; 55
rekord zárolás; 309, 311
rekord-módosítás; 54
relációs műveletek; 12
REPLACE; 86, 253, 270
REPLICATE; 256
REPORT FORM TO PRINT; 95, 96
REPORT; 96, 275

RESUME; 270
RETRY; 318
RETURN(); 318
RLOCK(); 312
RUN; 262
SCEDIT; 277
SEEK; 89, 253, 254
SELECT; 251
SET; 281
SET CARRY; 87
SET CATALOG TO; 98
SET COLOR TO; 93
SET DELETED; 88
SET DELIMITER; 93
SET DEVICE TO PRINT; 93, 95
SET ENCRYPTION; 293
SET EXCLUSIVE OFF; 315
SET EXCLUSIVE ON; 315
SET FILTER; 90
SET FORMAT TO; 93
SET MARGIN TO; 95
SET PRINT ON/OFF; 95
SET PRINT; 95
SET PRINTER TO; 95
SET RELATION TO; 90
SET SAFETY; 88
SET VIEW TO; 90
SKIP; 254
SORT; 85, 253, 269
SPACE; 256
SQRT; 256
STORE; 257
STR; 96, 257
STUFF; 257
SUBSTR; 257
SUPERVISOR; 289
SYLK; 265
szerkesztés; 336, 337, 349
szerkesztett; 338
szerkesztő; 338, 339, 340, 341, 348
szerkezeti file; 18, 106
szöveg file; 19
sztring műveletek; 12
szűrő feltételek; 64
TEDIT; 262
teljes képernyős műveletek; 13
teljes körű (FULL) mező előjog; 294
törlési előjog; 294
TOTAL; 270
TYPE; 95
újraélesztés; 318
UNISTAL; 288
UNLOCK; 312
UPDATE; 860
UPDATE FROM; 254
USE ... EXCLUSIVE; 315, 333
USE; 254, 267
ütközés; 309
VIEW; 269
view-lock; 314
WP; 262
ZAP; 86
zárolás; 309
zárolási időtartam; 312

Irodalomjegyzék

Dr.Peter Albrecht
dBase III
Das Datenbanksystem für 16-bit-Computer Markt & Technik, München, 1985.

dBase III PLus And Local Area Networks
Ashton-Tate, 1986.

FoxBase +
User Manual
Fox Software, Inc., 1987.

Getting Started dBase III PLUS
Ashton-Tate, 1985.

Learning and Using dBase III PLUS
Ashton-Tate, 1985.

James Matrin
Számítógépes adatbázis-szervezés
Műszaki Könyvkiadó, 1981.

Networking with dBase III PLUS
Ashton-Tate, 1985.

Novell-Netware/286
Novell, Inc., 1986.

Novell-Netware/86
Novell, Inc., 1986.

Programming With dBase III PLUS
Ashton-Tate, 1985.

Runtime + , Runtime + Reference
Ashton-Tate, 1985.

Hivatkozott márkanevek

A *"CLIPPER"* a ***Nantucket Corp***. márkaneve

A *"NANTUCKET"* a ***Nantucket Corp***. bejegyzett márkaneve

A *"dBase"*, *"dBase III PLUS"* és az *"Ashton-Tate"* az ***Ashton-Tate, Inc***. márkaneve

A *"dBase II"* az ***Ashton-Tate, Inc***. bejegyzett márkaneve

A *"Foxbase+"* a ***Fox Software, Inc***. márkaneve

Az *"IBM"*, a *"PC-DOS"* és az *"AT"* az ***International Business Machines Corp***. bejegyzett márkaneve

A *"Microsoft"* a ***Microsoft Corp***. bejegyzett márkaneve

Az *"MS-DOS"* és a *"MASM"* a ***Microsoft Corp*** márkaneve

A *"PC/XT"* az ***International Business Machines Corp***. márkaneve

A *"PLINK86-PLUS"* a ***Phoenix Technologies, Ltd***. márkaneve

Commodore 64 – IBM PC adatátvitel

Az általunk kifejlesztett intrface és adatátviteli programok segítségével Commodore 64 számítógépét intelligens adatrögzítő gépnek használhatja, amelynek floppy lemezéről az adatok átvihetők az IBM PC/XT/AT (vagy PROPER-16, Commodore PC 10, PC 20 stb.) számítógépre. Így az adatgyűjtés az IBM géptől elkülönítve történhet, egy vagy több Commodore 64-en, az IBM PC gépidejének lekötése nélkül. Az adatcsere kétirányú lehet; tehát az IBM gépről is küldhetünk át adatfile-okat a Commodore 64-re.
Az átviteli programok gyorsak (assembly nyelven íródtak), ellenőrzik és jelzik az átvitel során esetleg fellépő hibákat.

Az adatátviteli hardver és szoftver ára ÁFA-val: 28.500.-Ft

Érdeklődni lehet:
LSI Alkalmazástechnikai Tanácsadó Szolgálat
Software osztály (Budapest, X. Noszlopy u. 1.)
Telefon: 573-374, 570-433/382, /384, /386

COMMODORE 64 ÉS IBM PC ADATÁTVITEL

Az általunk kifejlesztett interface és adatátviteli programok segítségével *Commodore 64* számítógépét intelligens adatrögzítő gépnek használhatja, amelynek floppy lemezéről az adatok átvihetők az *IBM PC/XT/AT* (vagy *PROPER-16*, *Commodore PC 10/PC 20* stb.) számítógépre. Így az adatgyűjtés az *IBM* géptől elkülönítve történik, egy vagy több *Commodore 64*-en, az *IBM PC* gépidejének lekötése nélkül. Az adatcsere kétirányú lehet: tehát az *IBM* gépről is küldhetünk át adat-file-okat a *Commodore 64*-re.

Az adatátviteli programok gyorsak (assembly nyelven íródtak) Ellenőrzik és jelzik az átvitel során esetleg fellépő hibákat.

Az adatátviteli hardwer és software ára ÁFA-val:28.500.-Ft

Érdeklődni lehet: LSI Alkalmazástechnikai Tanácsadó Szolgálat
Software Osztály
Budapest,X.,Noszlopy u.1.VIII.825.
573-374 VAGY 570-433/382,384

Ennek a hirdetésnek a szövegét a *különféle formátumú magyar karakterkészlet* segítségével írtuk. A betűtípusok közül néhány:

- abcdefg...z 123...0 '~!@#$%^&*()-_=+[{]};:'"/?.>,< ABCD...Z
- aábcdeéfghií...z 123..0 '~=+"%$!/&_:-.;,? AÁBCDEÉFG...Z
- aábcd...z 123...0 '~+="%$!/& :-.;,? AÁBCDEÉFG...Z
- *aábcd...z 2345..o '~=+"%£!/&_:-;.?, AÁBCDEÉFG...Z*
- **aábcd...z 1234...0 '~+="%$!/&_:-.;,? AÁBCDEÉFG...Z**
- AÁBCD...Z123...0`~!@#$%^&*(.>,< AÁBCDEÉFG...Z
- "*†‡- -"ƒ¢●£¶¥™®§▶©
- | ‖ ▌█ ▓ ▒ ░ ≡ ⋰ ▄ – ▬ ┆ ‖ ▌ ≡ ⋱ ▀ … = ╓ ╥ ╖ ╕ ↖ ↑ ↗ ...stb.

A karakterkészlet ára ÁFA-val:600.-Ft/db

MEGRENDELEM AZ LSI ATSz KIADVÁNYAI KÖZÜL AZ ALÁBBIAKAT:

.... **db REXLIB** + melléklet
(Képernyő- és file-kezelő rezidens rutin és utility csomag MS-DOS és NOVELL Netware 286 környezetben (Turbo Pascal 4.0 alatt is) fejlesztők számára) 205.-Ft
.... db Orosz Judit: **IBM PC-HÁLÓZATOK ÁTTEKINTÉSE**
(LAN RENDSZEREK: Novell, NetWare, PC Network, 3Com3 +) 123.-Ft
.... db Erdős-Rusznák: **IBM PC XT/AT INFORMÁCIÓS KÁRTYA** 126.-Ft
.... db Dr.Pajor G.: **IBM PC-RŐL KEZDŐ FELHASZNÁLÓKNAK I. A HARDVER** 66.-Ft
.... db Dr.Pajor G.: **IBM PC-RŐL KEZDŐ FELHASZNÁLÓKNAK II. A SZOFTVER** ... 70.-Ft
.... db Kápolnai A.-Geszti P.: **IBM PC BASIC FEJLESZTŐRENDSZER** 236.-Ft
.... db Pirkó J.: **3D PERSPEKTÍVIKUS GRAFIKA TURBO PASCAL-BAN** 149.-Ft
.... db Dr.Úry-Bernáth-Darvas-Orosz: **IBM PC DOS I-II-III.**
(3.30 verzióval bővített kiadás) 692.-Ft
.... db Dr.Úry László: **SYMPHONY I.** 160.-Ft
.... db Dr.Úry László: **SYMPHONY II.** 133.-Ft
.... db Dr.Barakonyi: **IBM PC TÁBLÁZATKEZELŐ RENDSZEREK**
(LOTUS 1-2-3, FRAMEWORK, CALCSTAR) 314.-Ft
.... db Ráth-Éltető: **PROGRAM- ÉS ADATVÉDELEM IBM PC-RE** 97.-Ft
.... db Sipos-Soltész: **SZÖVEGSZERKESZTŐK IBM PC-re I.**
(WORDSTAR, CHIWRITER) 248.-Ft
.... db Bartha Attila: **NORTON**
(UTILITEIS, INTEGRATOR, EDITOR, COMMANDER, GUIDE) 150.-Ft
.... db Pirkó J.: **TURBO PASCAL** kezdőknek - haladóknak **4.0 verzióig** 192.-Ft
.... db Dóra és trsai.: **AZ IBM PC BELSŐ FELÉPÍTÉSE** 399.-Ft
.... db Mondovics és trsai.: **DBASE III PLUS, FOXBASE + , CLIPPER** I-II.k.... Irányár 450.-Ft

.... db Dr.Kovács Magda: **ANGOL-MAGYAR MIKROELEKTRONIKAI SZÓTÁR** 197.-Ft
.... db Berkes - Balogh - Kovács: **A SZÁMÍTÓGÉPES TÁVKÖZLÉS TELEMATIKAI SZOLGÁLATAI: TELETEX, FAKSZIMILE, VIDEOTEX**
(A könyvet a klasszikus hírközlés területén dolgozó szakembereknek és azoknak az informatikával foglalkozó olvasóknak ajánljuk, akik a számítógépes információcsere szabványokon alapuló megvalósulási formái iránt érdeklődnek.) 463.-Ft
.... db Dr.Arz-Dr.Lipóth-Dr.Merksz: **ROBOTMANIPULÁTOROK** 350.-Ft
.... db Dr.Siegler András: **ROBOTIRÁNYÍTÁSI MODELLEK** 350.-Ft

Név: . . . . . .

Cím: . . . . . .

Dátum: . . . . . .

Levélcím: LSI ATSz, Budapest 1300 Postafiók 114

I.–II. kötet ára: 479,-Ft

dBase (FoxBase, Clipper) programok hatékonyabb és egyben könnyebb készítése: P R G L I S T

A programcsomag célja, hogy a dBase (FoxBase vagy Clipper) alatt futtatandó programok struktúrájának felépítését, követését megkönnyítse.

A programcsomag a következő modulokból áll:

1. A programstruktúra kialakításának támogatása.
   A forrásprogram parancsait struktúrába rendezi, azaz a DO WHILE, IF, CASE és FOR feltételek alá tartozó programsorokat három karakterrel beljebb rendezi.

2. A program struktúrájáról táblázatokat készít.
   Az első táblázat minden programsorra feltünteti, hogy a benne foglalt utasítás tartozik-e valamely feltétel alá, ha igen, melyek azok és hogyan, milyen szinten ágyazódnak egymásba. Tartalmazza továbbá, hogy mely programsoroknál adtunk meg kilépési utasítást vagy rutinhívást.

A program ára ÁFA-val: 2.800-Ft

Kérésére díjmentesen elküldjük a program demonstrációs változatát.

Érdeklődni lehet:
LSI Alkalmazástechnikai Tanácsadó Szolgálat
Software osztály (Budapest, X. Noszlopy u. 1.)
Telefon: 573-374, 570-433/382, /384, /386